

**Vysoká škola báňská – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky**

**Vizualizace hodnot parametrů rádiového kanálu systému  
GSM a UMTS**

**Visualization of Values of Parameters for Radio Channel in  
GSM and UMTS**

**2015**

**Bc. Libor Stebel**

## Zadání diplomové práce

Student:

**Bc. Libor Stebel**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T059 Mobilní technologie

Téma:

Vizualizace hodnot parametrů rádiového kanálu systému GSM a UMTS  
Visualization of Values of Parameters for Radio Channel in GSM and UMTS

Zásady pro vypracování:

1. Proveďte komplexní rešerši parametrů tzv. Measurement Report pro systémy GSM a UMTS.
2. Navrhněte a realizujte webovou službu, která bude automaticky vizualizovat neměřené parametry rádiové kanálu systému GSM a UMTS. Součástí služby je ukládání hodnot z poskytnutého textového souboru do databáze a statistické vyhodnocení.
3. Proveďte testování služby a umístěte ji na veřejný www portál.

Seznam doporučené odborné literatury:

[1] EBERSPÄCHER, J, Hans-Jörg VÖGEL a Christian BETTSTETTER. GSM: switching, services, and protocols. 2nd ed. New York: Wiley, c2001, xiv, 332 p. ISBN 04-714-9903-X.

[2] EDITED BY HARRI HOLMA, Antti Toskala, Hans-Jörg VÖGEL a Christian BETTSTETTER. WCDMA for UMTS HSPA evolution and LTE: switching, services, and protocols. 5th ed. Chichester, West Sussex: Wiley, 2010, xiv, 332 p. ISBN 978-111-9991-908.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



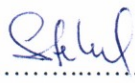
  
doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry

  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 30. dubna 2015

  
.....  
podpis studenta

## **Poděkování**

Rád bych poděkoval Ing. Liboru Michálkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

## **Abstrakt**

Cílem teoretické části této diplomové práce je popsat měření parametrů v systémech GSM a UMTS, v praktické části pak vytvořit automatizovaný systém pro vizualizaci těchto parametrů. Tento systém má za úkol číst soubory s naměřenými hodnotami, zpracovat je do vhodného formátu a poslat na server, kde se tyto hodnoty uloží do databáze. Další část této diplomové práce načte uložená data z databáze a vykreslí je v grafech a tabulkách na webové stránce. Součástí webové stránky bude i zobrazení blízké nebo vzdálené historie naměřených dat a výsledků z vybraných statistických metod. Kompletní webová aplikace bude zobrazovat naměřené hodnoty ze čtyř měřících stanic a bude veřejně dostupná pro kohokoliv.

## **Klíčová slova**

GSM; UMTS; Measurement Report; Java; MySQL; uložená procedura; webová služba; JAX-WS; SOAP; Apache Tomcat; Servlet; Google Charts; XHTML; JavaScript;

## **Abstract**

The aim of the theoretical part of this master thesis is to describe parameter measurement in GSM and UMTS systems. Then in the practical part create an automated system for visualization of these parameters. This system has to be able to read the file with measured values, convert them to a suitable format and send them to server, where these values are stored in a database. Another part of this master thesis retrieves stored data from the database and draws them in charts and tables on a web site. This web site also displays near and far history of measured values and the results of selected statistical methods. The complete web application will display measured values of the four measuring stations and will be publicly available to anyone.

## **Key words**

GSM; UMTS; Measurement Report; Java; MySQL; Stored procedure; Web service; JAX-WS; SOAP; Apache Tomcat; Servlet; Google Charts; XHTML; JavaScript;

## Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
<b>API</b>	Application Programming Interface	Aplikační programové rozhraní
<b>ASP.NET</b>	Active Server Pages .NET	Aktivní serverové stránky .NET
<b>BA</b>	BCCH Allocation	Seznam BCCH frekvencí sousedních buněk
<b>BCC</b>	Base Transceiver Station Color Code	Barevný kód základnové stanice
<b>BCH</b>	Broadcast Channel	Vysílací kanál
<b>BCCH</b>	Broadcast Control Channel	Vysílací řídicí kanál
<b>BER</b>	Bit Error Rate	Poměr chybných bitů
<b>BLER</b>	Block Error Rate	Poměr chybných bloků
<b>BSC</b>	Base Station Controller	Řídící jednotka základnových stanic
<b>BSIC</b>	Base Transceiver Station Identity Code	Identifikační kód základnové stanice
<b>BSS</b>	Base Station Subsystem	Subsystem základnových stanic
<b>BTS</b>	Base Transceiver Station	Základnová stanice
<b>CB</b>	Cell broadcast	Buňkové vysílání
<b>CC</b>	Country Code	Kód země
<b>CCPCH</b>	Common Control Physical Channel	Běžný řídicí fyzický kanál
<b>CI</b>	Cell Identifier	Identifikátor buňky
<b>CPCH</b>	Common Packet Channel	Běžný paketový kanál
<b>CPICH</b>	Common Pilot Channel	Běžný pilotní kanál
<b>CPU</b>	Central Processing Unit	Procesor
<b>CRC</b>	Cyclic Redundancy Check	Cyklický redundantní součet
<b>CSS</b>	Cascading Style Sheets	Kaskádové styly
<b>CSV</b>	Comma-Separated Values	Hodnoty oddělené čárkou
<b>DCCH</b>	Dedicated Control Channel	Vyhrazený řídicí kanál
<b>DCH</b>	Dedicated Channel	Vyhrazený kanál

---

<b>DPCCH</b>	Dedicated Physical Control Channel	Vyhrazený fyzický řídicí kanál
<b>DRAM</b>	Dynamic Random Access Memory	Dynamická paměť s náhodným přístupem
<b>EDO</b>	Extended Data Out	Rozšířený datový výstup
<b>FACCH</b>	Fast Associated Control Channel	Rychlý přidružený řídicí kanál
<b>FACH</b>	Forward Access Channel	Dopředný přístupový kanál
<b>FCCH</b>	Frequency Correction Channel	Kanál pro korekci frekvence
<b>FDD</b>	Frequency Division Duplex	Kmitočtově dělený duplex
<b>GPRS</b>	General Packet Radio Services	Paketově orientovaná datová služba
<b>GSM</b>	Global System for Mobile Communications	Globální systém pro mobilní komunikaci
<b>HDD</b>	Hard Disk Drive	Pevný disk
<b>HTML</b>	Hypertext Markup Language	Hypertextový značkovací jazyk
<b>HTTP</b>	Hypertext Transfer Protocol	Protokol pro přenos hypertextových dokumentů
<b>ID</b>	Identification	Identifikace
<b>IMSI</b>	International Mobile Subscriber Identity	Mezinárodní identita zákazníka
<b>IP</b>	Internet Protocol	Internetový protokol
<b>ISS</b>	Internet Information Services	Webový server od Microsoftu
<b>JAX-WS</b>	Java API for XML Web Services	Knihovna pro webové služby v jazyce Java
<b>JDBC</b>	Java Database Connectivity	Knihovna určená pro konektivitu s databází v jazyce Java
<b>JSF</b>	Java Server Faces	Serverové stránky jazyka Java
<b>JSON</b>	JavaScript Object Notation	JavaScriptový objektový zápis
<b>LA</b>	Location Area	Lokační území
<b>LAC</b>	Location Area Code	Kód území
<b>LAI</b>	Location Area Identity	Identifikátor území
<b>MNC</b>	Mobile Network Code	Kód mobilní sítě

---



<b>MS</b>	Mobile Station	Mobilní stanice
<b>MSC</b>	Mobile Switching Center	Centrum pro přepínání uzlů
<b>NAS</b>	Non Acces Stratum	Nepřístupná vrstva
<b>NCC</b>	Network Colour Code	Barevný kód sítě
<b>OS</b>	Operation System	Operační systém
<b>PCH</b>	Paging Channel	Kanál pro paging
<b>PLMN</b>	Public Land Mobile Network	Veřejná pozemní mobilní síť
<b>PSC</b>	Primary Scrambling Code	Primární kódovací kód
<b>P-CPICH</b>	Primary Common Pilot Channel	Primární běžný pilotní kanál
<b>P-TMSI</b>	Primary Temporary Mobile Subscriber Identity	Primární dočasná identita mobilního zákazníka
<b>RACH</b>	Random Access Channel	Náhodný přístupový kanál
<b>RAM</b>	Random Access Memory	Paměť s náhodným přístupem
<b>RLC</b>	Radio Link Control	Řízení rádiového spojení
<b>RNC</b>	Radio Network Controller	Řídící jednotka rádiové sítě
<b>RRC</b>	Radio Resource Control	Řízení rádiových zdrojů
<b>RSCP</b>	Received Signal Code Power	Obdržený výkon jednoho kódu
<b>RSSI</b>	Received Signal Strenght Indicator	Indikátor obdržené síly signálu
<b>RXLEV</b>	Received Signal Level	Úroveň přijatého signálu
<b>RXQUAL</b>	Received Signal Quality	Kvalita přijatého signálu
<b>SACCH</b>	Slow Associated Control Channel	Pomalý přidružený řídicí kanál
<b>SCH</b>	Synchronization Channel	Synchronizační kanál
<b>SDCCH</b>	Standalone Dedicated Control Channel	Samostatný vyhrazený řídicí kanál
<b>SFN</b>	System Frame Number	Číslo systémového rámce
<b>SIB</b>	System Information Block	Blok systémové informace
<b>SIM</b>	Subscriber Identity Module	Karta identifikující zákazníka
<b>SIR</b>	Signal-to-Interference Ratio	Poměr signálu k rušení
<b>SOAP</b>	Simple Object Access Protocol	Protokol pro výměnu XML zpráv
<b>SVG</b>	Scalable Vector Graphics	Škálovatelná vektorová grafika

<b>S-CPICH</b>	Secondary Common Pilot Channel	Sekundární běžný pilotní kanál
<b>TCH</b>	Traffic Channel	Provozní kanál
<b>TDD</b>	Time Division Duplex	Časově dělený duplex
<b>TDMA</b>	Time Division Multiple Access	Časově dělený mnohonásobný přístup
<b>TMSI</b>	Temporary Mobile Subscriber Identity	Dočasná identita mobilního zákazníka
<b>UE</b>	User Equipment	Uživatelské zařízení
<b>UMTS</b>	Universal Mobile Telecommunications System	Universální systém pro mobilní telekomunikace
<b>URA</b>	UTRAN Registration Area	Registrační území UTRAN
<b>URL</b>	Uniform Resource Locator	Jednotná adresa zdroje
<b>USIM</b>	UMTS Subscriber Identity Module	Karta identifikující zákazníka v UMTS síti
<b>UTRAN</b>	UMTS Terrestrial Radio Access Network	Pozemní rádiově přístupná síť UMTS
<b>VŠB - TUO</b>		Vysoká škola báňská - Technická univerzita Ostrava
<b>WCDMA</b>	Wideband Code Division Multiple Access	Širokopásmový vícenásobný přístup s kódovým dělením
<b>WSDL</b>	Web Services Description Language	Jazyk popisující webovou službu
<b>XHTML</b>	Extensible Hypertext Markup Language	Rozšířitelný hypertextový značkovací jazyk
<b>XML</b>	Extensible Markup Language	Rozšířitelný značkovací jazyk
<b>2G</b>	Second Generation	Mobilní síť 2. generace
<b>3G</b>	Third Generation	Mobilní síť 3. generace
<b>3GPP</b>	The 3rd Generation Partnership Project	Partnerský projekt třetí generace

# Obsah

Úvod.....	- 1 -
1 Měření v GSM.....	- 2 -
1.1 Prvky GSM sítě .....	- 2 -
1.2 Identifikátory .....	- 2 -
1.3 Vybrané kanály .....	- 3 -
1.4 Radio Subsystem Link Control .....	- 4 -
1.4.1 Měření rádiového kanálu.....	- 4 -
1.4.2 Nečinný režim (Idle mode).....	- 5 -
1.4.3 Režim spojení (Connected mode) .....	- 6 -
1.5 Regulace přenosového výkonu.....	- 7 -
2 Měření v UMTS .....	- 8 -
2.1 Prvky UMTS sítě.....	- 8 -
2.2 Měřené parametry fyzické vrstvy .....	- 8 -
2.3 Vybrané kanály .....	- 9 -
2.4 Radio Resource Control Protocol.....	- 9 -
2.4.1 Servisní stavy RRC (RRC Service States) .....	- 9 -
2.4.2 Nečinný režim (Idle mode).....	- 10 -
2.4.3 Režim spojení (Connected mode) .....	- 10 -
2.5 Řízení měření a podávání zpráv o měření .....	- 11 -
2.5.1 Řízení měření .....	- 11 -
2.5.2 Podávání zpráv o měření .....	- 12 -
2.6 Procedury měření handoveru.....	- 13 -
2.6.1 Intra-Mode Handover .....	- 13 -
2.6.2 Inter-Mode Handover .....	- 13 -
2.6.3 Inter-System Handover.....	- 13 -
2.7 Řízení výkonu .....	- 13 -
3 Sběr naměřených dat do souboru .....	- 15 -
4 Programové zpracování dat.....	- 17 -
4.1 Parsování dat .....	- 17 -

4.2	Poslání dat na server.....	- 17 -
4.3	Ošetření chybových stavů .....	- 19 -
4.3.1	Soubor chybí nebo nelze otevřít .....	- 19 -
4.3.2	Žádná naměřená data.....	- 19 -
4.3.3	Výpadek meteorologické stanice.....	- 19 -
4.3.4	Výpadek Telit modemu .....	- 19 -
4.3.5	Výpadek serveru.....	- 19 -
4.3.6	Výpadek měřicí stanice .....	- 20 -
4.4	Zpětné parsování dat .....	- 20 -
5	Zpracování na serveru .....	- 21 -
5.1	Virtuální stroj .....	- 21 -
5.2	Server Apache Tomcat 8 .....	- 21 -
5.2.1	Požadavky na server .....	- 21 -
5.2.2	Nastavení serveru .....	- 22 -
5.2.3	Spouštění serveru .....	- 22 -
5.2.4	Zastavování serveru.....	- 22 -
5.3	Webová služba .....	- 23 -
5.3.1	Rozhraní IDataLogger .....	- 23 -
5.3.2	Třída DataLogger .....	- 24 -
5.4	Logování chybových zpráv .....	- 24 -
6	Databáze MySQL.....	- 25 -
6.1	Požadavky na databázi .....	- 25 -
6.2	Nastavení MySQL.....	- 25 -
6.3	Tabulky .....	- 25 -
6.4	Procedury .....	- 28 -
6.5	Události .....	- 30 -
6.6	Databázová vrstva .....	- 30 -
7	Vizualizace dat.....	- 32 -
7.1	Servlety .....	- 32 -
7.2	Google Charts.....	- 33 -
7.3	XHTML stránky .....	- 35 -

7.4	Kalendář pro výběr data .....	- 39 -
7.5	Stylování.....	- 39 -
7.6	Logo webové aplikace.....	- 39 -
8	Testování.....	- 40 -
	Závěr .....	- 41 -
	Použitá literatura a zdroje.....	- 42 -
	Přílohy .....	- 43 -

## Úvod

K dnešnímu životu neodmyslitelně patří volání za pomoci mobilních telefonů. Dnes nejrozšířenější systém pro mobilní komunikaci je systém GSM (Global System for Mobile Communications), který se označuje jako systém 2. generace. Jako následovník GSM byl navržen systém UMTS (Universal Mobile Telecommunications System) označovaný jako systém 3. generace. UMTS však zcela nenahradil systém GSM, nýbrž byl implementován do jeho existující sítě a rozšiřuje tak GSM o nové služby (např. rychlejší přenos dat).

Během klidového stavu, ale i hovoru, si běžný uživatel není vědom toho, který z těchto uvedených systémů zrovna jeho zařízení používá. Neuvědomí si ani situaci, kdy zařízení během hovoru přepne z jednoho systému do druhého. Současná zařízení dokáží samostatně a plynule přepínat mezi GSM a UMTS. Tato funkcionalita by nebyla možná bez měření určitých parametrů rádiového spojení. U obou systémů je nejdůležitější zejména měření úrovně přijatého signálu ze základnové stanice. Měří se však mnoho dalších parametrů, které určují, zda je současná kvalita a síla signálu dostačující pro hovor nebo zda je potřeba přepnout do jiné buňky, popřípadě do jiného systému. Měření parametrů může provádět krom mobilního zařízení i základnová stanice. Na některých mobilních zařízeních lze tyto měřené parametry zobrazit v tzv. servisním menu. Tyto zařízení jsou vyhledávány jak servisními techniky, tak i nadšenci zabývajícími se důkladněji těmito systémy.

V této diplomové práci jsou první dvě kapitoly věnované právě měření příslušných parametrů systémů GSM a UMTS, které se měří za účelem řízení spojení mobilního zařízení se základnovou stanicí nebo za účelem přepínání do jiné buňky nebo systému. Následující kapitoly popisují tvorbu automatizovaného systému pro vizualizaci těchto měřených parametrů. Nejprve je uvedeno, jak se tyto data sbírají z modemů a meteorologických stanic do souborů. Poté je zmíněno, jak se tyto soubory programově zpracovávají a zpracované data posílají na server. Následuje kapitola popisující způsob uložení naměřených dat v databázi a kapitola o vizualizaci těchto dat ve webové aplikaci. Vlastní vypracování diplomové práce je zakončené otestováním vytvořeného automatizovaného systému.

# 1 Měření v GSM

Před samotným popisem principu měření parametrů v GSM je třeba popsat některé prvky sítě, identifikátory a kanály související s tímto měřením. Popis všech částí GSM není součástí této diplomové práce, proto se v případě hlubšího zájmu lze více dozvědět z [1][2][3].

## 1.1 Prvky GSM sítě

Následuje popis vybraných prvků GSM sítě.

- *Mobile Station* (MS) - mobilní stanice.
- *Base Transceiver Station* (BTS) - základnová stanice. Poskytuje rádiové kanály pro signalizaci a uživatelský provoz dat v buňce.
- *Base Station Controller* (BSC) - řídicí jednotka pro několik BTS. Řídí a kontroluje některé procesy BTS jako např. handover.
- *Base Station Subsystem* (BSS) - subsystém skládající se z několika BTS a BSC.
- *Mobile Switching Center* (MSC) - přepínací uzel GSM PLMN (Public Land Mobile Network) sítě. Poskytuje veškeré přepínací funkce uzlů fixní sítě.

## 1.2 Identifikátory

Následuje popis vybraných identifikátorů vyskytujících se v GSM.

- *Location Area Identity* (LAI) - strukturovaný, hierarchicky a mezinárodně jedinečný identifikátor každé oblasti v PLMN síti. Skládá se z mezinárodně standardizované části a z části závislé na operátorovi:
  - *Country Code* (CC) - kód země (3 číslice).
  - *Mobile Network Code* (MNC) - kód sítě (2 číslice).
  - *Location Area Code* (LAC) - kód území, který má maximálně 5 číslic nebo maximálně dvakrát 8 bitů zakódovaných hexadecimálně ( $LAC < FFFF_{hex}$ ).

LAI je vysílán pravidelně ze základnové stanice pomocí BCCH (Broadcast Control Channel) kanálu. Každá buňka je identifikována jednoznačně na rádiovém kanále jako patřící do určitého LA (Location Area) území a každá MS může stanovit její současnou lokaci pomocí LAI.

- *Cell Identifier* (CI) - jednotlivé buňky uvnitř LA jsou jednoznačně identifikovány pomocí CI, který je maximálně dvakrát 8 bitů. CI společně s LAI tvoří tzv. „*Global Cell Identifier*“ (LAI + CI), který zaručuje celosvětovou jedinečnost buňky.
- *Base Transceiver Station Identity Code* (BSIC) - za účelem rozlišit sousední základnové stanice, obdržují základní stanice BSIC (Base Transceiver Station Identity Code) kód, který se skládá ze dvou komponent:
  - *Network Color Code* (NCC) - barevný kód uvnitř PLMN (3 bity).

- *Base Transceiver Station Color Code* (BCC)- barevný kód základnové stanice (3 bity).

BSIC je vysílán periodicky pomocí základnové stanice na kanálech BCH (Broadcast Channel) a SCH (Synchronization Channel). Přímě přilehlá PLMN musí mít různé barevné kódy.

### 1.3 Vybrané kanály

Následuje popis vybraných kanálů v GSM.

- *Traffic Channel* (TCH) - provozní kanál. Slouží pro přenos uživatelsky užitečných dat (hovor, fax, data).
- *Broadcast Channel* (BCH) - jednosměrné vysílací kanály používané v BSS k vysílání stejné informace všem MS v buňce. Dělí se na tři kanály:
  - *Broadcast Control Channel* (BCCH) - tento kanál slouží pro vysílání sérií informačních zpráv směrem k MS. Tyto zprávy poskytují konfiguraci rádiového kanálu (jak už současně používané buňky, tak i sousedních buněk), synchronizační informace (jak frekvence, tak i číslování rámců) a registrační identifikátory (LAI, CI, BSIC). BCCH kanál je vysílán na první přiřazené frekvenci dané buňky (tzv. BCCH nosná frekvence).
  - *Frequency Correction Channel* (FCCH) - tento kanál vysílá informace pro kmitočtovou korekci směrem k MS.
  - *Synchronization Channel* (SCH) - tento kanál vysílá informace k identifikaci BTS (např. BSIC). Navíc také vysílá data pro rámcovou synchronizaci MS.
- *Dedicated Control Channel* (DCCH) - obousměrný signalizační kanál typu bod-bod. Zahrnuje následující kanály:
  - *Stand-alone Dedicated Control Channel* (SDCCH) - vyhrazený signalizační kanál typu bod-bod, který není vázán na existenci TCH kanálu. Je použit například pro signalizaci mezi MS a BSS pokud není aktivní spojení.
  - *Slow Associated Control Channel* (SACCH) - tento kanál je vždy přidružený a používán s kanály TCH nebo SDCCH. SACCH kanál nese informace pro optimální provoz rádiového spojení, jako např. pokyny pro synchronizaci a řízení vysílacího výkonu nebo zprávy o měření kanálu. Když zde nejsou k poslání žádná signalizační data, MS posílá zprávu o měření (tzv. „*Measurement Report*“) spolu s aktuálními výsledky z průběžně prováděného měření úrovně rádiového signálu.
  - *Fast Associated Control Channel* (FACCH) - tento kanál je přidělen pouze ve spojení s TCH kanálem a je určen pro krátkodobé použití další signalizace.

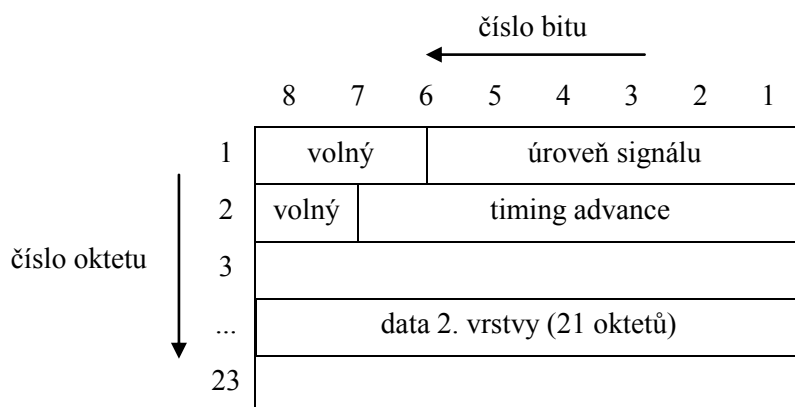


## 1.4 Radio Subsystem Link Control

Jedna z funkcí rádiového rozhraní je řízení rádiového spojení v originálním názvu „*Radio Subsystem Link Control*“. Hlavní aktivity této funkce jsou měření kvality přijatého signálu za účelem výběru buňky a přepínání mezi nimi (handover), dále pak kontrola výkonu vysílače.

Pokud není aktivní spojení, BSS nevykonává žádné úkoly. Nicméně MS neustále sleduje BCCH kanál současných a sousedních buněk. Tím je schopná vybrat nejvhodnější buňku pro komunikaci.

Během spojení (TCH nebo SDCCH) slouží funkce měření kanálu a řízení výkonu k udržení a optimalizaci rádiového kanálu. Tyto funkce pro kontrolu spojení jsou vykonávány na SACCH kanálu. Pro tyto účely jsou v SACCH bloku (viz obrázek 1.1) definována dvě pole, úroveň signálu a tzv. „*timing advance*“. Při downlinku tyto pole obsahují hodnoty přidělené od BSS. Při uplinku vloží MS její současně použité hodnoty. Hodnoty z měření kvality jsou vysílány v datové části SACCH bloku.



Obrázek 1.1: SACCH blok [1]

Řízení rádiového spojení může být rozděleno do třech úkolů: sběr a zpracování naměřených dat, řízení výkonu vysílače a řízení handoveru. Tyto úkoly se opakují ve smyčce každých 480 ms a jsou ukončeny pouze při ztrátě spojení. V každém kroku se počítají střední hodnoty dat. Nejprve jsou tato naměřená data poskytnuta pro kontrolu vysílacího výkonu MS a BSS. Poté jsou použita spolu s výsledkem z kontroly vysílacího výkonu pro proces handoveru.

### 1.4.1 Měření rádiového kanálu

Funkce „*Radio Subsystem Link Control*“ v MS zahrnuje identifikaci dosažitelné základnové stanice spolu s měřením jejich patřičných přijímaných úrovní signálu, a dále měření kvality kanálu. V nečinném režimu (tzv. „*idle mode*“) tato měření slouží k výběru současné základnové stanice.

Během spojení (např. u TCH nebo SDCCH s patřičným SACCH/FACCH kanálem) jsou tato naměřená data vysílána skrze SACCH kanál směrem k základnové stanici jako

tzv. „*Measurement Report/Measurement Info*“. Tyto zprávy slouží jako vstup pro algoritmy pro handover a řízení výkonu.

Měřenými objekty jsou na jedné straně uplink a downlink současného kanálu (TCH nebo SDCCH) a na druhé straně BCCH nosná, která nepřetržitě vysílá konstantním výkonem ze všech BTS ve všech časových slotech. Je zvláště důležité uchovávat vysílaný výkon BCCH nosné konstantní, aby bylo možné porovnávat sousední základnové stanice. Seznam frekvencí BCCH nosných sousedních základnových stanic značený BA (BCCH Allocation) je poskytován každé MS pomocí její současné BTS za účelem povolení měření všech buněk, které jsou kandidáty pro handover. Identita buňky (CI) je vysílána jako BSIC skrze BCCH kanál. Mimoto až 36 BCCH nosných frekvencí a jejich BSIC kódy mohou být uloženy na SIM kartě.

V principu GSM používá dva parametry k popisu kvality kanálu: RXLEV (Received Signal Level) neboli úroveň přijatého signálu měřená v dBm a RXQUAL (Received Signal Quality) neboli kvalita přijatého signálu měřená jako poměr chybných bitů BER (Bit Error Rate) v procentech před korekcí chyb. Síla přijatého signálu je měřena nepřetržitě MS a BTS v každém přijatém burstu v rozmezí od -110 dBm ( $= 10^{-14}$  W) do -48 dBm ( $= 15,8 \cdot 10^{-9}$  W). Požadované hodnoty RXLEV jsou získány průměrováním.

### 1.4.2 Nečinný režim (Idle mode)

Během nečinnosti si musí být MS vždy vědoma svého okolí. MS musí být schopná přiřadit se k takové buňce, od které spolehlivě zvládne dekodovat BCCH nosnou frekvenci. V tomto případě dokáže MS číst systémové a paging informace. Pokud se objeví požadavek na sestavení spojení, MS je s největší pravděpodobností připravena komunikovat se sítí.

Existují dvě možné počáteční situace:

- MS nemá žádné vědomosti o síti, zvláště o tom, které BCCH nosné frekvence jsou používány.
- MS má uložený seznam BCCH nosných.

V prvním méně příznivém případě z těchto dvou uvedených musí MS prohledávat skrze všech 124 GSM frekvencí, měřit jejich výkonovou úroveň signálu a spočítat průměr z alespoň 5 měření. Měření jednotlivých nosných frekvencí by mělo být rovnoměrně rozloženo v intervalu 3-5 sekund. Po nejvýše 5 sekundách je dostupných minimálně 629 naměřených hodnot, které umožňují určit 124 hodnot RXLEV. Nosné frekvence s nejvyšší hodnotou RXLEV jsou s největší pravděpodobností BCCH nosné frekvence. Finální identifikace nastává s burstem pro kmitočtovou korekci na kanále FCCH. Pokud byly obdrženy BCCH nosné frekvence již jednou nalezeny, MS se začne synchronizovat s každou z nich a začne číst systémové informace, počínajíc BCCH s nejvyšší RXLEV hodnotou.

Tento proces může být značně urychlen, pokud je seznam BCCH nosných frekvencí uložen na SIM kartě. V tomto případě se MS zkusí zesynchronizovat nejprve s některou známou BCCH nosnou ze seznamu. Pouze pokud nemůže najít žádnou z uložených BCCH nosných

frekvencí, začne provádět normální hledání popsané v předchozím případě. MS může ukládat několik seznamů pro několik nedávno navštívených sítí.

### 1.4.3 Režim spojení (Connected mode)

Během spojení za provozu (TCH) nebo signalizace (SDCCH) se měření kanálu MS objevuje v SACCH intervalu, který zahrnuje 104 TDMA rámců v případě TCH kanálu nebo 102 TDMA rámců v případě SDCCH kanálu.

Na straně kanálu jsou rozpoznávány dva parametry: úroveň přijatého signálu RXLEV a kvalita signálu RXQUAL. Tyto dvě hodnoty jsou průměrovány v každém SACCH intervalu (délka intervalu je 480 nebo 470.8 ms) a vysílány směrem k základnové stanici skrze SACCH kanál jako tzv. „*Measurement Report/Measurement Info*“. Tímto způsobem může být kanál, který je přidělen k MS, hodnocen z hlediska kvality downlinku. K tomuto měření downlinku pomocí MS navíc základnová stanice také měří RXLEV a RXQUAL hodnoty příslušného uplinku.

Pro rozhodnutí, zda se provede handover, musí být dostupné informace o možných cílových BTS pro handover. Pro tyto účely musí MS nepřetržitě sledovat BCCH nosné frekvence až šesti sousedních základnových stanic. Měření RXLEV sousedních BCCH nosných frekvencí je prováděno pomocí nepoužitých časových slotů MS. Výsledky měření šesti nejsilnějších BCCH signálů jsou zahrnuty ve zprávě „*Measurement Report*“ vysílané do BSS.

Nicméně samotné obdržení výkonové úrovně signálu a BCCH nosné frekvence nejsou dostatečná kritéria pro úspěšný handover. Kvůli opakovanému používání frekvencí v buňkových sítích, zvláště v případě malých clusterů, je možné, že buňka může přijímat stejnou BCCH nosnou frekvenci od více než jedné sousední buňky. Proto je nezbytné také znát identitu každé sousední buňky (BSIC). Zároveň s měřením úrovně signálu se MS musí synchronizovat s každou ze šesti sousedních BCCH nosných frekvencí a číst alespoň informace ze SCH kanálu.

Pro tento účel se musí nejprve najít FCCH burst z BCCH nosné frekvence, poté lze nalézt SCH v dalším TDMA rámcí. Jelikož FCCH/SCH/BCCH je vždy vysílán v nultém časovém slotu BCCH nosné frekvence, lze hledání FCCH kanálu během konverzace provádět pouze v nepoužitých rámcích. Tyto rámce jsou známy také jako vyhledávací rámce (v originále tzv. „*search frames*“).

Proto jsou zde přesně čtyři vyhledávací rámce v rámci SACCH intervalu o délce 480 ms. MS musí prozkoumat okolní BCCH nosné frekvence pro FCCH bursty za účelem synchronizace s nimi a dekodování SCH kanálu. Během operace synchronizace lze hledat tyto synchronizační body také uvnitř rámců. Je to možné díky aktuálnímu provoznímu kanálu a díky tomu, že příslušné BCCH nosné frekvence používají různé formáty multirámců. Zatímco provozní kanál používá 26 rámcový formát multirámce, nultý časový slot od BCCH nosné frekvence s FCCH/SCH/BCCH má 51 rámcový formát multirámce. Tento poměr různých multirámcových formátů má takový efekt, že relativní pozice vyhledávacího rámce (26. rámeček v TCH multirámci) je posunuta vzhledem k BCCH multirámci o přesně jeden rámeček každých

240 ms. Obrazně řečeno, vyhledávací rámec cestuje podél BCCH multirámce tak, že nanejvýš po 11 TCH multirámcích (= 1320 ms) se burst pro kmitočtovou korekci sousedních buněk stane viditelným ve vyhledávacím rámci.

Tímto způsobem je MS schopna určit BSIC pro příslušné RXLEV naměřené hodnoty. Pouze ta měření BCCH nosných frekvencí, jejichž identita může být stanovana bez jakýchkoliv pochyb, jsou zahrnuta do zpráv „*Measurement Report*“ posílaných základnové stanici. Základnová stanice nyní může rozhodnout o handoveru na základě těchto hodnot, vzdálenosti MS a momentální interferenci nepoužitých časových slotů.

Algoritmy pro rozhodování o handoveru nejsou zahrnuty v GSM standardu. Síťoví operátoři mohou používat algoritmy, které jsou optimalizované pro jejich síť nebo lokální situaci. GSM pouze dává základní návrh, který splňuje minimální požadavky pro algoritmus rozhodující o handoveru. Tento algoritmus definuje prahové hodnoty, které musí být porušeny v jednom nebo druhém směru, aby se dospělo k bezpečnému rozhodnutí o handoveru a k zabránění takzvaných ping-pong handoverům, které oscilují mezi dvěma buňkami.

## 1.5 Regulace přenosového výkonu

Řízení výkonu GSM vysílače má za úkol omezovat vysílací výkon MS na nejnižší potřebnou úroveň tak, aby základnová stanice získávala signál z různých MS s přibližně stejnou výkonovou úrovní. Pro tyto účely je definováno 16 kontrolních výkonových stupňů. Stupeň 0 (43 dBm = 20 W) až stupeň 15 (13 dBm = 0,02 W). Základnová stanice může zvyšovat vysílací výkon MS v krocích po 2 dBm (= 1,58 mW) od nejnižšího stupně 15 až po maximální výkon příslušné výkonové třídy. Podobně vysílací výkon základnové stanice může být také kontrolován v krocích po 2 dBm s výjimkou BCCH nosné frekvence, která musí zůstat konstatní, aby umožňovala srovnávací měření sousedních BCCH nosných frekvencí pomocí MS.

Řízení vysílacího výkonu je založeno na měření hodnot RXLEV a RXQUAL, pro které jsou definované horní a dolní prahové hodnoty pro uplink a downlink. Pokud prahová hodnota uplinku je překročena, vysílací výkon MS je redukován. V opačném případě, pokud je signál pod prahovou hodnotou, MS zvýší svůj vysílací výkon. Stejným způsobem to platí pro vysílací výkon základnové stanice, který může být upraven, pokud kritéria pro downlink překročí prahovou hodnotu v jednom ze směrů.

Dokonce pokud úroveň signálu mobilní nebo základnové stanice zůstává mezi danými prahovými hodnotami, současné RXLEV/RXQUAL hodnoty mohou způsobit změnu na jiný kanál stejné nebo jiné buňky, a to kvůli prahovým hodnotám handoveru. Z tohoto důvodu je kontrola vysílací prahové hodnoty neprodleně následovaná kontrolou prahové hodnoty handoveru jako druhá část „*Radio Subsystem Link Control*“. Pokud je jedna z prahových hodnot překročena v jakémkoliv směru a vysílací výkon nemůže být upraven odpovídajícím způsobem (např. příslušný vysílací výkon již dosáhl minimální nebo maximální hodnoty), je to příčina pro handover, kterou BSS musí okamžitě oznámit do MSC. [1][2][3]

## 2 Měření v UMTS

Stejně jako tomu bylo v kapitole 1, je i zde nutné pospat jednotlivé pojmy související s měření v UMTS. Popis všech pojmů není součástí této diplomové práce a v případě zájmu lze více zjistit v [4][5][6].

### 2.1 Prvky UMTS sítě

Následuje popis vybraných prvků UMTS sítě.

- *User Equipment* (UE) - uživatelské zařízení skládající se z mobilního zařízení a USIM.
- *UMTS Terrestrial Radio Access Network* (UTRAN) - skupina všech síťových prvků UMTS sítě. Skládá se ze dvou hlavních prvků a to NodeB a RNC.
- *NodeB* - základnová stanice komunikující s UE. Poskytuje rádiové rozhraní a vykonává některé základní operace při komunikaci s UE.
- *Radio Network Controller* (RNC) - prvek sítě zodpovědný za řízení rádiových zdrojů UTRAN sítě. Řídí jednu nebo více NodeB.

### 2.2 Měřené parametry fyzické vrstvy

3GPP standardy specifikují určité měření závislé na fyzické vrstvě, které mohou být prováděny jak pomocí UE, tak i pomocí UTRAN, a posílány vyšším vrstvám. S hodnocením výkonu rádiového spojení a jeho analýzou je nejvíce spojené měření následujících parametrů:

- *Received Signal Code Power* (RSCP) - přijatý výkon jednoho kódu určený z pilotních symbolů měřených na P-CPICH (Primary Common Pilot Channel) kanále. Při downlinku může být měření RSCP použito pro výpočet ztrát odčítáním hodnot vysílaného výkonu na P-CPICH kanálu. Výkon vysílaný na P-CPICH je znám operátorovi a může být také vyčten ze systémové informace vysílané skrze BCCH (Broadcast Control Channel) transportní kanál.
- *Received Signal Strength Indicator* (RSSI) - udává celkový výkon obdržený od všech NodeB skrze celou šířku 5 MHz pásma. RSSI zahrnuje i tepelný šum a šum generovaný přijímačem.
- *Ec/No* - udává poměr obdržené energie na jeden chip v P-CPICH kanálu vůči celkové energii v celém 5 MHz pásmu. Ec/No je stejný jako RSCP/RSSI. Měření Ec/No je velice přesné a patří mezi nejdůležitější měření UE ve WCDMA (Wideband Code Division Multiple Access) pro plánování a optimalizaci sítě. Ec/No se také používá pro handover.
- *Signal-to-interference ratio* (SIR) - tento parameter se měří za účelem řízení výkonu UE. Určuje ho UE z pilotních bitů DPCH (Dedicated Physical Control Channel) kanálu.

- *Block Error Rate* (BLER) - udává odhad poměru chybných bloků transportního kanálu. Hlavní důvod měření BLER je dát zpětnou vazbu pro řízení výkonu při nastavování cílového SIR.

## 2.3 Vybrané kanály

Následuje popis vybraných kanálů v UMTS.

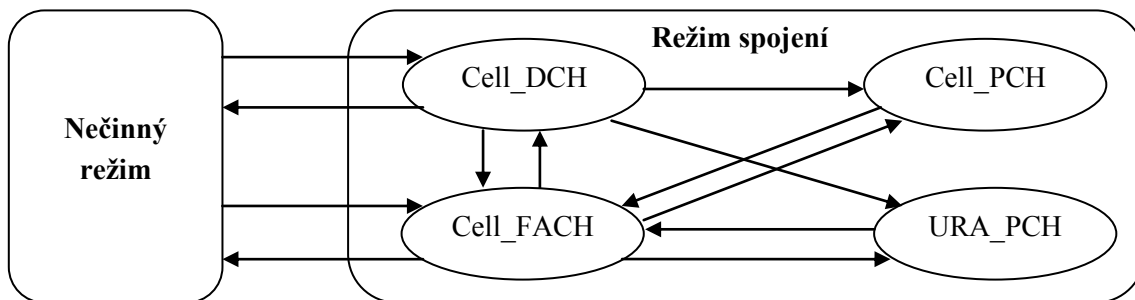
- *Broadcasting Control Channel* (BCCH) - downlink kanál pro vysílání systémových řídicích informací.
- *Dedicated Control Channel* (DCCH) - obousměrný kanál typu bod-bod, který přenáší vyhrazené kontrolní informace mezi UE a RNC. Tento kanál je zřízen během procedury sestavení RRC (Radio Resource Control) spojení.
- *Common Pilot Channel* (CPICH) - nemodulovaný kódovaný kanál, který pro šifrování používá specifický PSC (Primary Scrambling Code) buňky. Úkolem CPICH kanálu je pomáhat UE určit kanálový odhad pro DCH (Dedicated Channel) nebo poskytnout odkaz na kanálový odhad běžným kanálům, které nejsou přidružené k DCH kanálu. Ani když je k CPICH kanálu namapovaný nějaký transportní kanál, nenese žádnou informaci z vyšších vrstev. Existují dva typy CPICH - primární a sekundární. Primární CPICH (P-CPICH) má vždy PSC a fixně přidělený kanálový kód a pro buňku nebo sektor existuje takový kanál pouze jeden. Sekundární CPICH (S-CPICH) může mít jakýkoliv kanálový kód o délce 256 a také sekundární scrambling kód. P-CPICH kanál je důležitý pro měření za účelem výběru buňky a handoveru.
- *Synchronization Channel* (SCH) - kanál používán pro vyhledávání buňek. Stejně jako u CPICH kanálu není ani na SCH kanál namapovaný žádný jiný transportní kanál. SCH kanál se rozlišuje na dva kanály a to primární (P-SCH) a sekundární (S-SCH). P-SCH a S-SCH jsou posílány paralelně.

## 2.4 Radio Resource Control Protocol

Hlavní části pro řízení signalizace mezi UE a UTRAN jsou RRC zprávy. RRC zprávy přenášejí všechny parametry potřebné pro nastavení, modifikaci a uvolnění protokolových entit 1. a 2. vrstvy. RRC zprávy v sobě také nesou veškerou signalizaci vyšších vrstev.

### 2.4.1 Servisní stavy RRC (RRC Service States)

UE se může nacházet ve dvou základních režimech: nečinný režim („*idle mode*“) a režim spojení („*connected mode*“). Režim *connected* může být dále rozdělen na servisní stavy, které definují, jaký druh fyzických kanálů UE používá. Mobilita UE během režimu „*connected*“ je řízená RRC signalizací (měření, handover, aktualizace buňky apod.). Jednotlivé stavy a spojení mezi nimi lze vidět na obrázku 1.2.



Obrázek 1.2: Stavy UE a spojení mezi nimi [4]

### 2.4.2 Nečinnný režim (Idle mode)

V „idle“ režimu UE po zapnutí vybírá (buď automaticky nebo manuálně) PLMN, aby ji mohl kontaktovat. Následně UE hledá vhodnou buňku vybrané PLMN sítě, vybere buňku poskytující dostupné služby a naladí se na její řídicí kanál. Tento výběr je znám jako „*Camping on a cell*“, po kterém je UE schopné přijímat systémové informace a CB (Cell Broadcast) zprávy. UE setrvává v nečinnném režimu, dokud nevyšle požadavek na sestavení RRC spojení. V nečinnném režimu je UE identifikováno pomocí NAS (Non Access Stratum) identit, jako jsou IMSI (International Mobile Subscriber Identity), TMSI (Temporary Mobile Subscriber Identity) a P-TMSI (Primary TMSI). UTRAN kromě toho nemá informace o jednotlivých UE v nečinnném režimu a může pouze adresovat všechny UE v buňce nebo UE monitorující výskyt pagingu.

### 2.4.3 Režim spojení (Connected mode)

V Cell\_DCH stavu je k UE přidělen vyhrazený fyzický kanál a UE je známý jeho obsluhujícímu RNC na buňkové úrovni nebo na úrovni aktivního setu. UE provádí podle řídicích informací obdržených od RNC měření kanálu a posílá zpět zprávy o měření (tzv. „*Measurement Report*“).

V Cell\_FACH stavu není k UE přidělen žádný vyhrazený fyzický kanál, ale místo toho jsou pro vysílání signalizačních zpráv a malého množství uživatelských dat použity RACH (Random Access Channel) a FACH (Forward Access Channel) kanály. UE je v tomto stavu taky schopné naslouchat BCH kanálu za účelem získávání systémových informací. Při vyžádání z UTRAN může být použitý také CPCH (Common Packet Channel) kanál. V tomto stavu UE provádí opakovaný výběr buňky, po kterém následuje poslání zprávy „*Cell Update*“ do RNC. Tímto způsobem RNC zjistí lokaci UE na buňkové úrovni. Pokud nová buňka patří do jiného systému (např. GPRS), tak se UE přepne do nečinnného režimu a používá přístupové metody daného systému.

V Cell\_PCH stavu je UE stále známý jeho obsluhujícímu RNC na buňkové úrovni, ale může být dosažen pouze skrze PCH (Paging Channel) kanál. V tomto stavu je spotřeba baterie UE menší než u Cell\_FACH stavu. UE také naslouchá systémovým informacím z BCH kanálu. Pokud UE provede nový výběr buňky, přepne se samostatně do Cell\_FACH stavu kvůli tomu, aby mohl provést proceduru „*Cell Update*“. Poté se znova vrátí do stavu Cell\_PCH.

Pokud nová buňka patří do jiného systému, tak se UE přepne do nečinného režimu a používá přístupové metody daného systému.

URA\_PCH stav je velice podobný stavu Cell\_PCH s výjimkou toho, že UE po každém novém výběru buňky neprovádí proceduru „*Cell Update*“, ale místo toho čte identity URA (UTRAN Registration Area) z BCH kanálu a informuje RNC o jeho lokaci pouze při změně URA. Změna se provádí pomocí procedury „*URA Update*“, která je podobná „*Cell Update*“ proceduře (UE se vždy přepne do Cell\_FACH stavu, aby mohlo provést „*URA Update*“ a následně zpět do stavu URA\_PCH).

UE opouští režim spojení a vrací se do nečinného režimu poté, co RRC spojení je zrušeno nebo selže.

## 2.5 Řízení měření a podávání zpráv o měření

Měření prováděné pomocí UE jsou řízené pomocí RNC za použití RRC zpráv, které udávají co měřit, kdy měřit a jak podat zprávu o měření. RRC signalizace je také použita při podávání zpráv o naměřených hodnotách z UE směrem do UTRAN.

### 2.5.1 Řízení měření

Metody řízení měření a oznamování výsledků jsou navrženy tak, aby byly velice flexibilní. RNC může spustit, zastavit nebo změnit počet paralelních měření v UE a každé z těchto měření (zahrnující i způsob podání zprávy) může být řízené nezávisle na ostatních. Informace o řízení měření je zahrnuta v SIB (System Information Block) typu 12 a SIB typu 11. Pro UE v Cell\_DCH stavu může být použita také jednoúčelová zpráva „*Measurement Control*“.

Informace o řízení měření obsahují:

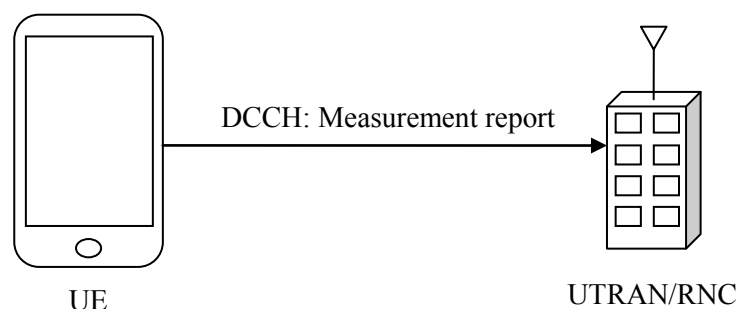
- *Measurement identity number* - identifikační číslo měření. Referenční číslo použité v UTRAN při modifikaci nebo uvolnění měření a v UE při posílání zpráv o měření.
- *Measurement command* - povel k měření. Může být buď nastavování, modifikace nebo uvolnění.
- *Measurement type* - typ měření. Jeden ze sedmi typů z předdefinovaného seznamu, kde každý typ popisuje co má UE měřit. Jsou definovány jako:
  - Intra-frekvenční měření: měření na downlink fyzických kanálech na stejné frekvenci jako aktivní set.
  - Inter-frekvenční měření: měření na downlink fyzických kanálech na frekvencích, které se liší od frekvence aktivního setu.
  - Inter-systémové měření: měření na downlink fyzických kanálech příslušajících jinému rádiově přístupovému systému než UTRAN, např. GSM.
  - Měření objemu provozu: měření provozního objemu při uplinku.
  - Měření kvality: měření parametrů kvality, např. BLER na downlink transportním kanálu.
  - Interní měření: měření přenosového výkonu a úrovně přijatého signálu z UE.



- Měření lokačních služeb: základní měření prováděné pomocí UE pro síťově založenou metodu pozicování.
- *Measurement objects* - objekty, které UE musí měřit, a jejich odpovídající informace. Při měření handoveru potřebuje UE informace o buňce, aby mohla provádět měření na určitých intra-frekvenčních, inter-frekvenčních nebo inter-systémových buňkách. Při měření objemu provozu obsahuje tento parametr identifikaci transportního kanálu.
- *Measurement quantity* - množství, které UE měří.
- *Measurement reporting quantities* - množství obsažené ve zprávách od UE.
- *Measurement reporting criteria* - kritérium spouštějící podávání zprávy o měření (periodicky nebo závisle na nějaké události).
- *Reporting mode* - specifikuje, jestli UE vysílá zprávy o měření pomocí potvrzovaných nebo nepotvrzovaných datkových přenosů z RLC (Radio Link Control).

### 2.5.2 Podávání zpráv o měření

Procedura podávající zprávy o měření (viz obrázek 1.3) je zahájena na straně UE poté, co jsou splněna všechna kritéria pro podání zprávy o měření. UE posílá zprávu „*Measurement Report*“ zahrnující identifikační číslo zprávy a výsledek měření.



Obrázek 1.3: *Procedura podání zprávy o měření [4]*

Zpráva „*Measurement Report*“ je použita v Cell\_DCH a Cell\_FACH stavech. Ve stavu Cell\_FACH slouží pouze pro podání zprávy o měření objemu provozu. Měření objemu provozu může být spuštěno také ve stavech Cell\_PCH a URA\_PCH, ale nejprve musí UE přepnout do stavu Cell\_FACH, než je schopna poslat zprávu o měření. Aby bylo možné přijmout informaci o měření v době, kdy se zřizuje vyhrazený fyzický kanál, může UTRAN požádat UE o připojení zpráv o měření k následujícím zprávám:

- *RRC Connection Request* - zpráva poslaná za účelem sestavení RRC spojení.
- *Initial Direct Transfer* a *Uplink Direct Transfer* zprávy.
- *Cell Update* zpráva.
- *Measurement Report* zpráva poslaná k ohlášení objemu provozu v Cell\_FACH stavu.

## 2.6 Procedury měření handoveru

V UTRAN se objevují tyto handovery: „*Intra-Mode Handover*“, „*Inter-Mode Handover*“ a „*Inter-System Handover*“. „*Intra-Mode Handover*“ může být měkký, měkkčí nebo tvrdý. „*Inter-Mode Handover*“ je handover do režimu TDD (Time Division Duplex). „*Inter-System Handover*“ je handover do GSM.

### 2.6.1 Intra-Mode Handover

Tento typ handoveru závisí na měření  $E_c/N_0$  prováděném skrze CPICH kanál. Definované parametry, které mohou být měřeny pomocí UE skrze CPICH, jsou RSCP, RSSI a  $E_c/N_0$ .

Existují také další parametry, které mohou být použity pro rozhodnutí o handoveru v UTRAN, avšak současné algoritmy rozhodující o handoveru zůstávají otázkou realizace. Jeden z těchto parametrů je SIR, který udává informaci o ortogonalitě buňky a je měřený ve všech případech za účelem řízení výkonu.

Dodatečná zásadní informace pro měkký handover je relativní načasování informace mezi dvěma buňkami. Pokud jsou buňky uvnitř oken o délce 10 ms, relativní načasování může být získáno z fáze PSC, neboť použitá perioda kódu je 10 ms. Pokud je načasování větší, pak je UE nucen dekódovat SFN (System Frame Number) z primárního CCPCH (Common Control Physical Channel) kanálu. To vždy zabere nějaký čas a také může obsahovat chyby, což vyžaduje CRC (Cyclic Redundancy Check) kontrolu u SFN. Pokud je informace o načasování k dispozici ze seznamu buněk sousedů, nemá okno o délce 10 ms význam. Pro tvrdý handover mezi frekvencemi není potřeba tak přesná informace o načasování. Získávání dalších měření je mírně komplikované, protože UE musí měřit na odlišné frekvenci.

### 2.6.2 Inter-Mode Handover

Dvourežimový UE operující ve FDD (Frequency Division Duplex) režimu měří při požadavku z UTRAN úroveň signálu TDD buněk dostupných v dané oblasti. Pro měření mohou být použity bursty TDD CCPCH kanálu poslané dvakrát během TDD rámce o délce 10 ms, neboť v downlinku je garantována jejich existence. TDD buňky ve stejné pokrývané oblasti jsou synchronizovány.

### 2.6.3 Inter-System Handover

Pro handover mezi UTRAN a GSM jsou stejné požadavky jako pro handover v rámci samotného GSM. UE normálně obdrží GSM SCH kanál v tzv. „*compressed mode*“ režimu a umožní měřit další frekvenci.

## 2.7 Řízení výkonu

Těsné a rychlé řízení výkonu je možná nejdůležitější aspekt ve WCDMA zejména při uplinku. Mobil s nadměrným výkonem by bez řízení výkonu mohl zablokovat celou buňku. Vyskytuje se zde tzv. „*near-far*“ problém.

Řešením pro řízení výkonu ve WCDMA je metoda nazvaná „*Fast Closed Loop Power Control*“. U tzv. „*Closed Loop Power Control*“ při uplinku provádí základnová stanice odhady přijímaného SIR a porovnává ho s cílovým SIR. Pokud je naměřený SIR větší než cílový SIR, vyzve základnová stanice mobilní stanici ke zmenšení výkonu a naopak při menší SIR ke zvětšení výkonu. Tento měřicí proces se opakuje 1500 krát za sekundu na každé mobilní stanici, a tím operuje rychleji, než může dojít k nějaké významné ztátě na trase. Tedy „*Closed Loop Power Control*“ předchází veškerým výkyvům výkonu mezi všemi uplink signály obdrženými v základnové stanici.

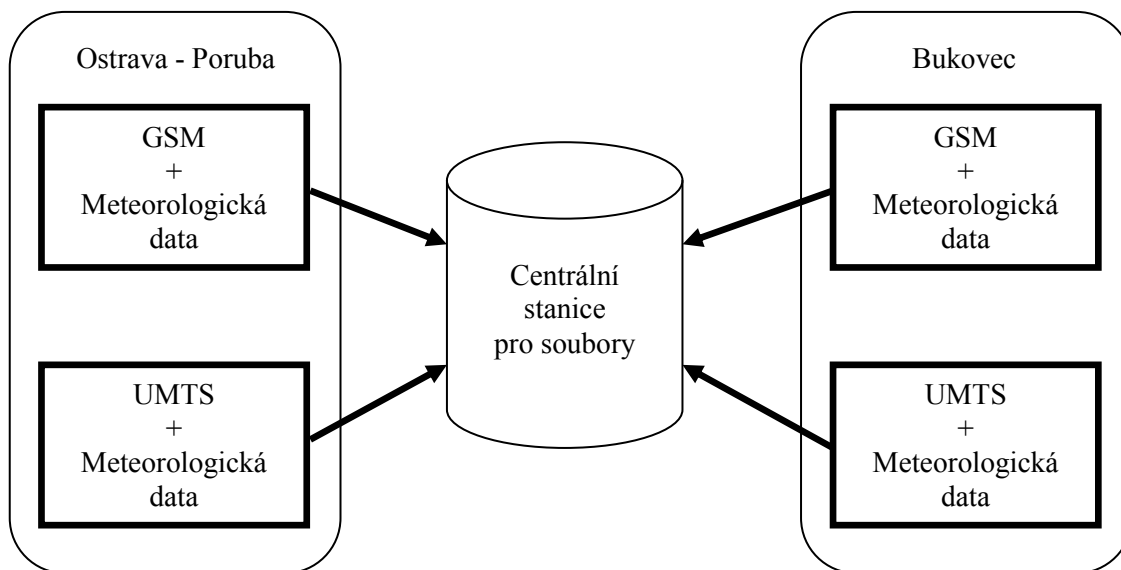
Stejná „*Closed Loop Power Control*“ technika je použita pro downlink, ačkoli zde je motivace jiná. Při downlinku žádný „*near-far*“ problém nenastává. Všechny signály uvnitř jedné buňky vychází z jedné základnové stanice do všech mobilních stanic. Je však žádoucí poskytnout malé množství přídavného výkonu mobilním stanicím na okraji buňky, protože zde trpí rušením z ostatních buněk.

„*Closed Loop Power Control*“ nařizuje mobilní stanici používat vysílací výkon úměrný převrácenému přijímacímu výkonu (nebo SIR). Tím docílí, že slábnoucí výkon kanálu se v podstatě vyrovná. Velmi vysoký vysílací výkon však může způsobit rušení v dalších buňkách.

Je zde ještě jedna metoda spojená s „*Closed Loop Power Control*“ a to „*Outer Loop Power Control*“. „*Outer Loop Power Control*“ přizpůsobuje cílový SIR potřebám individuálního rádiového spojení se zaměřením na konstantní kvalitu, obvykle definovanou jako určitý cílový BER nebo BLER. „*Outer Loop Power Control*“ je potřeba při uplinku i downlinku, protože v obojím se provádí „*Fast Loop Power Control*“. [4][5][6]

### 3 Sběr naměřených dat do souboru

Při navrhování automatizovaného systému pro vizualizaci naměřených parametrů nebylo cílem vytvořit od základu nové řešení, nýbrž byla vyžadována návaznost na již existující počáteční situaci, která datuje svůj počátek na rok 2011. V této počáteční situaci (viz obrázek 3.1) jsou k dispozici čtyři měřicí stanice, které nepřetržitě měří meteorologické hodnoty a hodnoty z mobilních rádiových sítí GSM nebo UMTS. Tyto hodnoty ukládají periodicky do logovacího souboru.

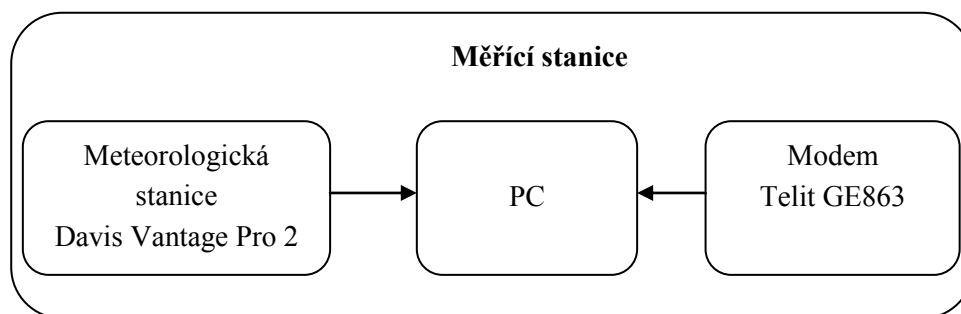


Obrázek 3.1: Počáteční rozmístění a spojení jednotlivých stanic

Dvě měřicí stanice se nacházejí v areálu Vysoké školy báňské - Technické univerzity Ostrava a dvě v obci Bukovec. Každá měřicí stanice je složena z klasického počítače s OS Linux, ke kterému jsou připojeny modem značky Telit a meteorologická stanice Davis Vantage Pro 2. Schéma zapojení každé měřicí stanice lze vidět na obrázku 3.2.

Modem Telit GE863 slouží pro získávání parametrů z mobilních rádiových systémů GSM nebo UMTS. U stanic měřících systém GSM se do logovacího souboru ukládají tyto hodnoty: BSIC, LAC, CI, ARFCN a úroveň signálu. U stanic měřících UMTS systém pak tyto hodnoty: stav hovoru, PSC, RSCP, Ec/No a režim mobilní stanice.

Z meteorologické stanice se do logovacího souboru ukládají tyto hodnoty: teplota, pocitová teplota, vlhkost, rosný bod, déšť, směr větru, rychlost větru, tlak, trend tlaku, východ slunce a západ slunce.



Obrázek 3.2: Schéma zapojení měřicí stanice

Meteorologická data a data z Telit modemu dohromady s hodnotami data a času tvoří každý řádek logovacího souboru (ukázka viz příklad 3.1). Tento soubor je ve formátu CSV a pro každý nový den se tvoří znova s novým unikátním názvem obsahující datum a název stanice. Začátkem nadcházejícího dne se tyto soubory kopírují z jednotlivých stanic na další, pro všechny společnou, centrální stanici.

---

```
1.01.2014;00:00;12;55;7050;A863;75;-78;56;7050;A865;77;-98;52;7050;
AEDD;107;-99;46;7050;A864;59;-100;FF;FFFF;0000;0;-111;FF;FFFF;0000;0;
-111;FF;FFFF;0000; 0;-111;1.7;92;0.5;NE;1.4;967.7;-1;0.5;1.7;0.0;
07:41;15:55
```

---

Příklad 3.1: Jeden řádek z logovacího souboru pro GSM síť

## 4 Programové zpracování dat

Jelikož jedním z požadavků pro výsledný systém bylo zobrazování co nejaktuálnějších dat na webové stránce, nepřicházela v úvahu manipulace s daty uloženými na centrální stanici, protože by jednoduše nebyly aktuální. Proto je potřeba zpracovávat logovací soubory již na jednotlivých měřicích stanicích, a to za současného zapisování do těchto souborů. Pro tento účel se tedy vytvořily programy v programovacím jazyce Java přizpůsobené pro konkrétní logovací soubory. Aby se programy mohly spouštět, bylo potřeba nainstalovat aktuální běhové prostředí jazyka Java na každou stanici.

### 4.1 Parsování dat

Tyto již zmíněné programy se spouští linuxovým nástrojem crontab každou minutu (viz příklad 4.1). Program pokaždé nejprve vytvoří kopii souboru, ze které se za normálních okolností přečtou všechna doposud nepřečtená data, tedy data za poslední minutu. Jelikož za tuto minutu může přibýt více než jeden řádek, udělá se z těchto řádků příslušný průměr. Zároveň se do paměti programu (tzv. „*Preferences*“) ukládá datum a čas posledního záznamu a počet již přečtených znaků. Tím se docílí přeskočení již zpracovaných dat. Na konci procesu parsování se vytvořená kopie smaže.

---

```
* * * * * /usr/bin/java -jar /home/SGSlogger/skripty-dp/Poruba2G.jar
```

---

Příklad 4.1: Příkaz v crontabu pro spuštění Java programu každou minutu

### 4.2 Poslání dat na server

Po úspěšném přečtení dat z bezchybného souboru má program k dispozici jeden nebo více zpracovaných řádků určených pro poslání. Program se připojuje k webové službě, která je vystavena na serveru, a posílá jí data (ukázka posílání dat viz příklad 4.2). Samotné data jsou ve skutečnosti serializovaná instance třídy s názvem `Data2GArray`, která v sobě nese naměřené hodnoty. Webová služba přenáší informace ve formě SOAP (Simple Object Access Protocol) zpráv, což není nic jiného, než spojení HTTP protokolu s XML souborem (ukázka SOAP zprávy viz příklad 4.3). [7]

---

```
URL wsdl = new URL("http://" + ipaddress + ":" + port + WSDL_PATH);
DataLoggerService ls = new DataLoggerService(wsdl);
IDataLogger service = ls.getDataLoggerPort();
Data2GArray array = new Data2GArray();
array.item = data;
ret = service.insertDataPorubaA2G(array);
```

---

Příklad 4.2: Volání konkrétní metody webové služby

Na straně serveru existuje úplně stejná třída `Data2GArray`, do které se zase přijatá data deserializují. Tímto se dostane na stranu serveru stejný objekt s naměřenými daty, jako je na straně měřicí stanice.

---

```
Payload: <?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
  <ns2:insertDataPorubaB3G xmlns:ns2="http://stebel.vsb.cz/">
    <dataArray>
      <item>
        <barometer>1023.3</barometer>
        <callStatus>true</callStatus>
        <dewPoint>-4.2</dewPoint>
        <ecio>-8.0</ecio>
        <falling>-1</falling>
        <heatIndex>-1.5</heatIndex>
        <humidity>81</humidity>
        <mode>0</mode>
        <psc>310</psc>
        <rain>0.0</rain>
        <rscp>-90</rscp>
        <sunrise>5:45</sunrise>
        <sunset>18:04</sunset>
        <temperature>-1.3</temperature>
        <thwIndex>-3.9</thwIndex>
        <time>20150323000316</time>
        <windChill>-3.7</windChill>
        <windDir>NNW</windDir>
        <windSpeed>1.3</windSpeed>
      </item>
      <item>...</item>
    </dataArray>
  </ns2:insertDataPorubaB3G>
</S:Body></S:Envelope>
```

---

**Příklad 4.3: Tělo SOAP zprávy s naměřenými daty**

## 4.3 Ošetření chybových stavů

Občas nastane situace, kdy se přihodí nějaká chyba a logování do souboru neběží podle představ. S těmito stavy však musí program pro zpracovávání souboru počítat.

### 4.3.1 Soubor chybí nebo nelze otevřít

První z chybových stavů nastává, pokud logovací soubor nelze programově otevřít nebo dokonce chybí. Stav, kdy soubor nejde otevřít, může být způsoben tím, že je otevřen jinou aplikací. V tomto případě program pro zpracování souboru počká 1 sekundu a pokusí se o to znova. Při desátém neúspěšném pokusu program skončí.

### 4.3.2 Žádná naměřená data

Další z chybových stavů nastává, když program po zpracování souboru nezíská žádná data k poslání na server. Tento stav může být způsoben výpadky meteorologické stanice a současně Telit modemem nebo i z jiných důvodů. Program tento stav řeší tím, že ani nekontaktuje server a jednoduše skončí. Při dalším spuštění pokračuje pouze s větším časovým rozmezím pro čtení, tedy snaží se přechít i rozmezí, které v předchozích spuštěních nenašel (data se mohla mezitím nahrát do logovacího souboru). Toto zvětšování časového rozmezí končí v případě nalezení následujících korektních dat.

### 4.3.3 Výpadek meteorologické stanice

V tomto stavu chybí v logovacím souboru všechny hodnoty z meteorologické stanice. Program tento stav rozpozná a pošle serveru pouze korektní data z mobilní sítě (pokud zde jsou). Meteorologická data pošle s chybovým kódem, díky kterému server zjistí, že tyto data chybí.

### 4.3.4 Výpadek Telit modemu

Při výpadku samotného Telit modemu chybí v logovacím souboru veškeré hodnoty o měření mobilní sítě. V tomto stavu program opět nekontaktuje server i přesto, že např. data z meteorologické stanice v souboru jsou. Jednoduše samotná meteorologická data nejsou žádoucí.

### 4.3.5 Výpadek serveru

Při výpadku na straně serveru neběží ani kontaktní webová služba obsahující metody pro příjem dat. Tento výpadek může být způsobený např. výpadkem proudu, plánovanou odstávkou nebo i z jiných důvodů. V tomto případě ukládá program na straně měřicí stanice zpracovaná data do fronty (další soubor ve formátu CSV). Při každém dalším spuštění nejprve kontroluje dostupnost serveru. Pokud je dostupný, tak mu nejdříve pošle data z fronty, frontu smaže a až poté pokračuje klasickým způsobem. Pokud není dostupný, tak přidá další zpracovaná data na konec fronty.



#### 4.3.6 Výpadek měřicí stanice

Nejméně žadaná, ale také nejméně běžná je situace, kdy vypadne celá měřicí stanice. To se může stát např. při výpadku proudu. V této situaci nefunguje přidávání hodnot do logovacího souboru. Po opětovném spuštění měřicí stanice začne program číst svým klasickým způsobem aktuální soubor podle současného data a času.

#### 4.4 Zpětné parsování dat

Jelikož v době tvorby této diplomové práce byly nashromážděny soubory s naměřenými daty za dobu až tři let, bylo potřeba vytvořit pomocné programy, které tyto data zpracují zpětně. Oproti již dříve zmíněným programům nepotřebují tyto pomocné programy posílat zpracovaná data nikam na server, nýbrž je ukládají přímo do databáze. Jsou vytvořeny tedy speciálně pro daný virtuální stroj. Pro každou měřicí stanici jsou vytvořeny vždy dva programy, jeden parsující soubory a druhý pro tvorbu průměrů. Vstupem do těchto programů je pouze počáteční a koncové datum, podle kterých zpracují příslušné soubory (viz příklad 4.4).

---

```
java -jar /home/data/ParserPoruba2G.jar 20150101 20150131
```

---

Příklad 4.4: Příkaz pro spuštění Java programu parsujícího soubory zpětně

## 5 Zpracování na serveru

Data ze všech měřících stanic se posílají na společný server, který je zpracuje a uloží do databáze. Pro příjem dat ze všech měřících stanic je na serveru vystavena webová služba, která je pro tyto účely přímo určená.

### 5.1 Virtuální stroj

Pro finální nasazení této diplomové práce je pomocí virtualizačního nástroje VMware vytvořen virtuální stroj ve strojovně budovy Fakulty elektrotechniky a informatiky sídlící v porubském areálu Vysoké školy báňské - Technické univerzity Ostrava. Jednotlivé parametry tohoto virtuálního stroje lze vidět v tabulce 5.1.

Processor	8 x Intel® Xeon® CPU E5-2630 v3 @ 2.40 GHz
RAM	32 GB EDO DRAM
HDD	34 GB
OS	Ubuntu 14.04.1 LTS

Tabulka 5.1: Parametry virtuálního stroje

### 5.2 Server Apache Tomcat 8

Na již zmíněném virtuálním stroji je rozběhnutý webový server Apache Tomcat 8. Apache Tomcat byl vybrán z mnoha dalších webových serverů kvůli jeho dostačujícím prostředkům pro realizaci této diplomové práce a také díky dalším doporučením.

#### 5.2.1 Požadavky na server

Na začátku této práce bylo specifikováno několik požadavků, které ovlivnily samotný výběr serveru. Původně bylo zamýšleno použití ISS (Internet Information Services) serveru od společnosti Microsoft a práci vytvářet pomocí technologie ASP.NET. Jeden z požadavků však určoval, že server musí běžet na stroji založeném na OS Linux. Tímto požadavkem tedy odpadla možnost použít ISS server a byl vybrán Apache Tomcat 8, který je multiplatformní. V návaznosti na výběr webového serveru bylo rozhodnuto i o programovacím jazyce pro serverovou část, tedy o jazyce Java. Tím vznikla nutnost nainstalovat běhové prostředí jazyka Java na virtuální stroj.

V jazyce Java existuje podobná alternativa k technologii ASP.NET, a to technologie JSF (Java Server Faces), která je již ve verzi JSF 2.2 (k datu 26.3.2015) [10]. Původně se zamýšlelo vytvořit webové stránky kompletně touto technologií, což se později ukázalo jako nevhodný výběr. V současné webové aplikaci je technologie JSF používána pouze pro jednu malou funkcionalitu. Ve finále jsou použity klasické XHTML stránky v kombinaci s Java Servlety a JavaScriptem. Apache Tomcat 8 tuto kombinaci zvládá bezproblémově.

### 5.2.2 Nastavení serveru

Jelikož je Apache Tomcat 8 po stažení ve výchozím nastavení, bylo zapotřebí upravit některé soubory a jejich hodnoty. Jako první byl upraven soubor *tomcat-users.xml*, do kterého bylo zapotřebí přidat administrátorské účty pro ovládání webového serveru přes webový prohlížeč (viz příklad 5.1).

---

```
<role rolename="manager-gui"/>
<user username="*****" password="*****" roles="manager-gui"/>
<role rolename="admin-gui"/>
<user username="*****" password="*****" roles="admin-gui"/>
```

---

Příklad 5.1: Administrátorské účty v souboru *tomcat-users.xml*

Dále byl přidán soubor *setenv.sh*, do kterého se zapisují paměťové nároky webového serveru. V základním nastavení používá webový server maximálně 128 MB paměti RAM. Při této hodnotě docházelo často k výpadku serveru z důvodu nedostatku paměti. Proto byla maximální hodnota využití RAM navýšena na 8 GB (viz příklad 5.2). Tento soubor čte Apache Tomcat automaticky při každém spuštění.

---

```
export JAVA_OPTS="-Dfile.encoding=UTF-8 -server -Xms8192m -Xmx8192m -
XX:PermSize=1024m -XX:MaxPermSize=1024m"
```

---

Příklad 5.2: Příkaz v souboru *setenv.sh* pro zvýšení paměťových nároků serveru

Posledním krokem v rámci nastavování serveru bylo mazání nadbytečných souborů. Apache Tomcat v základu poskytuje různé webové stránky s příklady a návody apod., které nejsou pro běh webové aplikace důležité.

### 5.2.3 Spouštění serveru

Po konfiguraci potřebných částí je server připraven ke spuštění. Spouští se jednoduchým skriptem (viz příklad 5.3), který je již obsažen v souborovém systému celého Apache Tomcat 8. Navíc byl do nástroje crontab přidán příkaz, aby se skript spouštěl automaticky při restartování virtuálního stroje.

---

```
sh /home/VSBRadioSignals/apache-tomcat-8.0.15/bin/startup.sh
```

---

Příklad 5.3: Příkaz pro spuštění webového serveru Apache Tomcat 8

### 5.2.4 Zastavování serveru

Stejně jako pro spuštění webového serveru, existuje i skript zastavující Apache Tomcat (viz příklad 5.4).

---

```
sh /home/VSBRadioSignals/apache-tomcat-8.0.15/bin/shutdown.sh
```

---

Příklad 5.4: Příkaz pro zastavení webového serveru Apache Tomcat 8

## 5.3 Webová služba

Při realizaci této diplomové práce bylo zapotřebí vybrat způsob, kterým si budou měřicí stanice a server předávat naměřená data. Pro tyto účely byly vybrány tzv. webové služby poskytující několik standardů pro posílání dat z jedné aplikace do druhé. V této aplikaci je konkrétně použita knihovna Apache CXF 2.7.1, určená pro programovací jazyk Java, která v sobě zároveň zahrnuje knihovnu JAX-WS (Java API for XML Web Services) poskytující nástroje a metody pro komunikaci skrze webové služby. [7][11]

### 5.3.1 Rozhraní IDataLogger

Konkrétní webová služba v této webové aplikaci je vytvořena pomocí rozhraní s názvem `IDataLogger`, které v sobě zahrnuje čtyři metody (viz příklad 5.5). Každá metoda je určená pro jinou měřicí stanici a obecně všechny slouží pro příjem naměřených dat. Třídy, jejichž objekty obsahují naměřené hodnoty, jsou shodné s třídami na straně měřicí stanice. O samotnou serializaci a deserializaci objektů do SOAP zpráv posílaných skrze HTTP protokol se stará již knihovna JAX-WS.

---

```
@WebService(name="IDataLogger", targetNamespace="http://stebel.vsb.cz/")
@SOAPBinding(style=Style.RPC)
public interface IDataLogger {

    @WebMethod(operationName = "insertDataPorubaA2G")
    public int insertDataPorubaA2G(@WebParam(name = "dataArray")
    ArrayList<Data2G> array);

    @WebMethod(operationName = "insertDataPorubaB3G")
    public int insertDataPorubaB3G(@WebParam(name = "dataArray")
    ArrayList<Data3G> array);

    @WebMethod(operationName = "insertDataBukovec2G")
    public int insertDataBukovec2G(@WebParam(name = "dataArray")
    ArrayList<Data2G> array);

    @WebMethod(operationName = "insertDataBukovec3G")
    public int insertDataBukovec3G(@WebParam(name = "dataArray")
    ArrayList<Data3G> array);

}
```

---

Příklad 5.5: Webová služba a její metody

Vystavení webové služby na web provádí samotný webový server při spouštění webové aplikace. Výsledkem vystavení je odkaz na WSDL (Web Services Description Language) soubor, který popisuje dané metody a IP adresu koncového bodu webové služby. Tento WSDL soubor je veřejný pro všechny a za normálních okolností si může kdokoli vytvořit příslušné klienty posílající data. V tomto případě je však do hlavičky POST zprávy HTTP protokolu přidáno uživatelské jméno a heslo a služba v případě nekorektních údajů odpoví chybovým kódem.[7]

### 5.3.2 Třída **DataLogger**

Tato třída je implementací rozhraní `IDataLogger` popsaného v předchozí kapitole. V případě příchozích dat nejprve ověřuje správnost jména a hesla. Pokud tyto údaje nesedí, zahodí přijatá data a odpoví chybovým kódem 401. V opačném případě pokračuje dále a zjišťuje, zda existují na serverové straně nějaká uložená data. Tato data vznikají v případě, kdy není dostupná databáze a je třeba uložit data dočasně do souboru, aby se o ně nepřišlo. Pokud taková data najde, snaží se je vložit do databáze. Poté následuje již vkládání nových dat. Pokud je vložení úspěšné, odpoví služba kódem 200. Při selhání uloží data do již zmíněného dočasného souboru na disku. Odpověď je opět 200, jelikož data jsou již na straně serveru, a program na straně měřicí stanice nemusí tato data opětovně posílat.

## 5.4 Logování chybových zpráv

Pro logování některých chybových zpráv do souboru je zde použita knihovna `log4j` ve verzi 2.1. Nastavení logování je uvedeno v souboru `log4j2.xml`, ve kterém se specifikuje cílový adresář pro uložení logu, intervaly pro tvorbu nových souborů a úroveň ukládaných zpráv. V této webové aplikaci se do logu ukládají pouze chybové zprávy z webové služby a z databázové vrstvy. Ukázku použití v programovacím jazyce Java lze vidět na příkladu 5.6.

---

```
Logger logger = LogManager.getLogger("WS_insert_data_to_PorubaA2G");  
logger.error("SQL Exception, discarding data");
```

---

Příklad 5.6: Ukázka použití knihovny `log4j` v jazyce Java

## 6 Databáze MySQL

Jak již bylo několikrát zmíněno v předchozích kapitolách, veškerá data z měřících stanic se ukládají do databáze. Pro řešení této práce byla zvolena databáze MySQL ve verzi 5.5 zejména kvůli počátečním požadavkům na celkový systém. Naměřená data se rozdělují do několika tabulek a jsou zpracovávána a optimalizována několika uloženými procedurami, o kterých se lze více dozvědět v kapitole 6.4.

### 6.1 Požadavky na databázi

Při výběru databáze sehrál největší roli fakt, že musí být bezplatná. Dalším kritériem bylo množství dat vkládajících se do databáze, které se pohybuje okolo 1,5 GB za rok a předpokládaná doba fungování se odhaduje na nejméně 10 let. V neposlední řadě byla důležitá spolehlivost a ověřenost databáze. Ze všech těchto kritérií vyplynula jako nejlepší volba databáze MySQL.

### 6.2 Nastavení MySQL

Databáze MySQL použita pro naše naměřená data je nainstalována na stejném virtuálním stroji jako webový server Apache Tomcat. Po její instalaci však byla v určitém počátečním nastavení, které plně nevyužívalo možnosti virtuálního stroje. Zejména byla schopna operovat pouze na jednom vlákne, což při používání 8 jádrového procesoru nebylo zcela efektivní. V konfiguračním souboru databáze tedy byly upraveny položky pro podporu vícevláknových procesů a taktéž byla navýšena velikost vyrovnávací paměti pro ukládání tabulek a indexovaných dat (viz příklad 6.1).

---

```
...  
innodb_buffer_pool_instances = 4  
innodb_buffer_pool_size = 4G  
innodb_io_capacity = 2000  
innodb_read_io_threads = 64  
innodb_thread_concurrency = 0  
innodb_write_io_threads = 64  
...
```

---

*Příklad 6.1: Upravené položky konfiguračního souboru databáze zvyšující její efektivitu*

### 6.3 Tabulky

Příchozí data z každé měřící stanice se rozdělují do několika tabulek. Z výkonnostních důvodů, ale také kvůli přehlednosti, se pro výsledný web nepoužívají pouze surová data, nýbrž i jejich hodinové a denní průměry. Taktéž se objevují některé výpočty, které jsou v daný okamžik pro všechny uživatele stejné. Tyto výpočty tedy stačí vypočítat pouze jednou

a uchovávat jejich výsledek v jiné tabulce. Uživatelé tímto způsobem dostanou rychlejší odpověď na webu.

#### Popis jednotlivých tabulek:

- PorubaA2G, PorubaB3G, Bukovec2G a Bukovec3G - obsahují veškerá surová data v takovém tvaru, v jakém byla přijatá od jednotlivých měřicích stanic. Obsahují tedy nejvíce řádků ze všech ostatních tabulek. Každý záznam těchto surových dat seskupuje dohromady hodnoty z měření mobilní sítě i z meteorologického měření (viz tabulka 6.1).

PorubaA2G		PorubaB3G	
Pole	Typ	Pole	Typ
id	int(11)	id	int(11)
datetime	bigint(20)	datetime	bigint(20)
bsic	smallint(6)	callstatus	tinyint(1)
lac	varchar(5)	psc	smallint(6)
cellid	varchar(5)	rscp	smallint(6)
arfcn	smallint(6)	ecio	float
power	smallint(6)	mode	smallint(6)
temperature	float	temperature	float
humidity	smallint(6)	humidity	smallint(6)
dewpoint	float	dewpoint	float
winddir	varchar(5)	winddir	varchar(5)
windspeed	float	windspeed	float
barometer	float	barometer	float
falling	smallint(6)	falling	smallint(6)
windchill	float	windchill	float
thwindex	float	thwindex	float
heatindex	float	heatindex	float
rain	float	rain	float
sunrise	varchar(5)	sunrise	varchar(5)
sunset	varchar(5)	sunset	varchar(5)

Tabulka 6.1: Formát jednotlivých polí tabulek PorubaA2G a PorubaB3G

Tabulky Bukovec2G a Bukovec3G jsou velice podobné tabulkám PorubaA2G a PorubaB3G (viz tabulka 6.1), pouze neobsahují pole thwindex.

Většina dotazů na tyto tabulky omezuje výběr podle pole datetime, které určuje časovou značku záznamu. Proto je k tomuto poli vytvořen index, který mnohonásobně zrychluje výsledné vyhledávání v tabulce.

- `PorubaA2G_1h`, `PorubaB3G_1h`, `Bukovec2G_1h` a `Bukovec3G_1h` - mají úplně stejný formát jako tabulky `PorubaA2G`, `PorubaB3G`, `Bukovec2G` a `Bukovec3G`. Obsahují však hodinové průměry naměřených dat. Tyto průměry počítají uložené procedury (viz kapitola 6.4).
- `PorubaA2G_1d`, `PorubaB3G_1d`, `Bukovec2G_1d` a `Bukovec3G_1d` - obsahují denní průměry naměřených dat, stejně jako tomu bylo u hodinových průměrů. Jejich formát je opět stejný jako u tabulek se surovými daty (viz tabulka 6.1) a opět je počítají uložené procedury (viz kapitola 6.4).
- `2GCurrent` a `3GCurrent` - obsahují výsledky různých hodnot vypočtených z parametrů mobilní sítě za poslední 1 hodinu, 12 hodin, 24 hodin, 1 týden, 1 měsíc a 1 rok. Tyto hodnoty jsou v daný okamžik pro všechny uživatele stejné a není třeba je vypočítávat pro každého z nich zvlášť. Hodnoty jsou vypočítávány uloženými procedurami každou minutu. Formát jednotlivých polí těchto tabulek lze vidět v tabulce 6.2, kde obě tabulky mají speciální pole `stationID`, které udává typ stanice a zároveň, zda se jedná o výpočty za 1 hodinu, 12 hodin atd.

2GCurrent		3GCurrent	
Pole	Typ	Pole	Typ
stationID	smallint(6)	stationID	smallint(6)
cellid	varchar(5)	psc	varchar(5)
lac	varchar(50)	rscpMax	smallint(6)
bsic	varchar(50)	rscpMin	smallint(6)
arfen	varchar(50)	rscpAvg	float
powerMax	smallint(6)	rscpMedian	float
powerMin	smallint(6)	rscpMode	smallint(6)
powerAvg	float	rscpDev	float
powerMedian	float	ecioMax	float
powerMode	smallint(6)	ecioMin	float
powerDev	float	ecioAvg	float
		ecioMedian	float
		ecioMode	float
		ecioDev	float

Tabulka 6.2: Formát jednotlivých polí tabulek `2GCurrent` a `3GCurrent`

- `WeatherCurrent` - je zde ze stejného důvodu jako byly tabulky `2GCurrent` a `3GCurrent`. Nyní se však uchovávají předvypočtené hodnoty meteorologických dat. Tato tabulka však neobsahuje všechny naměřené veličiny, nýbrž pouze ty, které se zobrazují na webové stránce. Jednoduše není třeba počítat nepoužívané veličiny. Opět je tvořena uloženými procedurami každou minutu. Formát jednotlivých polí této tabulky lze vidět v tabulce 6.3.



Pole	Typ
stationID	smallint(6)
function	varchar(10)
temperature	float
dewpoint	float
windchill	float
humidity	float
rain	float
barometer	float
windspeed	float

Tabulka 6.3: *Formát jednotlivých polí tabulky WeatherCurrent*

- `LastWeather` - uchovává poslední korektně naměřené meteorologické hodnoty. Je tvořena opět uloženou procedurou každou minutu a pokaždé obsahuje pouze 4 řádky. Formát jednotlivých polí této tabulky je uveden v tabulce 6.4.

Pole	Typ
station	varchar(10)
datetime	bigint(20)
temperature	float
dewpoint	float
windchill	float
humidity	smallint(6)
barometer	float
windspeed	float
winddir	varchar(5)

Tabulka 6.3: *Formát jednotlivých polí tabulky LastWeather*

## 6.4 Procedury

Z logických, ale hlavně i z výkonnostních důvodů je zřejmé, že se hodí některé úkony provádět přímo v databázi MySQL. Výpočet, který by bylo v programovacím jazyce Java třeba provést několika přístupy do databáze, lze jednodušeji a daleko rychleji provést přímo v databázi pomocí jedné uložené procedury. V jazyce Java se pak volá pouze tato uložená procedura. Všechny následně zmíněné procedury jsou vytvořeny zvlášť pro každou stanici a často se liší pouze v tabulce vstupních dat.

### Popis jednotlivých procedur:

- `average_hour` a `average_day` - vybírají hodnoty z tabulky surových dat a tvoří jejich hodinové, popřípadě denní průměry. Vstupem do těchto procedur je začátek a konec časového období pro výběr a nová časová značka příslušného

zprůměrovaného výsledku. Výsledky se ukládají do tabulek pro hodinové a denní průměry (viz kapitola 6.3).

- `allweather`, `1h_allweather` a `1d_allweather` - počítají vybrané statistické metody pro vybrané meteorologické hodnoty. Těmito metodami jsou maximum, minimum, prostřední hodnota, průměr, medián, modus a standardní odchylka. Rozdíl mezi těmito třemi procedurami je pouze v tom, z jaké tabulky se berou vstupní hodnoty (surová data, hodinový průměr, denní průměr).
- `allweather_current`, `1h_allweather_current` a `1d_allweather_current` - počítají stejné statistické metody, jako tomu bylo v předchozím bodě. Rozdíl je však v tom, že tento výsledek nevrací jako výstup procedury, nýbrž ho zapisují do tabulky `WeatherCurrent` (viz kapitola 6.3). Tyto procedury se spouští každou minutu a počítají tyto statistické metody vždy pro poslední 1 hodinu, 12 hodin, 24 hodin, 1 týden, 1 měsíc a 1 rok. Jednotlivé procedury se liší opět tím, z jaké tabulky se vybírají vstupní data (např. pro poslední 1 hodinu se berou surová data, ale pro poslední 1 týden už data z hodinových průměrů).
- `powertable`, `1h_powertable` a `1d_powertable` - se mírně liší pro stanice měřící 2G síť a 3G síť. V obojím případě se počítají vybrané statistické metody z naměřených hodnot mobilního signálu (úroveň signálu u 2G, RSCP a Ec/No u 3G). U 2G se však rozděluje podle identifikátoru CI a jsou k nim přidány ještě další identifikátory, jako LAC, BSIC a ARFCN. U 3G se rozděluje podle identifikátoru PSC. Tyto vybrané statistické metody jsou maximum, minimum, prostřední hodnota, průměr, medián, modus a standardní odchylka. Jednotlivé procedury se mezi sebou opět liší v tabulce vstupních dat.
- `powertable_current`, `1h_powertable_current` a `1d_powertable_current` - počítají stejné statistické metody, jako tomu bylo v předchozím bodě. Výsledek ovšem zapisují do tabulek `2GCurrent` a `3GCurrent` (viz kapitola 6.3). Tyto procedury se spouští každou minutu a počítají tyto statistické metody pro poslední 1 hodinu, 12 hodin, 24 hodin, 1 týden, 1 měsíc a 1 rok. Liší se mezi sebou opět v tabulce vstupních dat.
- `index_table_last_weather` - vybírá poslední korektní meteorologická data z tabulek `PorubaA2G`, `PorubaB3G`, `Bukovec2G` a `Bukovec3G` a ukládá je do tabulky `LastWeather`. Korektními daty se zde myslí ty, které jsou pro daný záznam vyplněné, tedy nenulové. Tato procedura se spouští každou minutu.
- `table_server_info` - je spíše informační. Počítá a vrací aktuální velikosti tabulek s naměřenými daty.

## 6.5 Události

Část z uložených procedur, popsanych v kapitole 6.4, je spouštěna opakovaně pomocí tzv. událostí (viz příklad 6.2). Pro tyto účely jsou vytvořeny tři takové události (`currentRefreshPower`, `currentRefreshWeather` a `indexRefresh`), které jsou spouštěny každou minutu. Rozdělené do tří událostí jsou pouze kvůli přehlednosti, ale z funkčního hlediska by stačila pouze jedna společná událost. Události je třeba v nastavení databáze povolit, jelikož v základu je jejich spouštění vypnuto. [8]

---

```
CREATE EVENT indexRefresh
ON SCHEDULE EVERY 1 MINUTE
DO CALL index_table_last_weather();
```

---

Příklad 6.2: *Událost `indexRefresh` spouštějící proceduru `index_table_last_weather`*

## 6.6 Databázová vrstva

Databázová vrstva slouží jako prostředník mezi webovou aplikací a databází MySQL. Díky JDBC (Java Database Connectivity) API umožňuje v programovacím jazyce Java volat dotazy a procedury z MySQL databáze. Obecný princip většiny metod v této vrstvě je sestavení spojení s databází, provedení dotazu nebo procedury a uzavření spojení.

### Popis jednotlivých tříd:

- `DBConnectionManager` - poskytuje metody pro sestavení a uzavření spojení s databází MySQL. Při sestavování spojení je výstupem metody s názvem `getConnection` instance třídy `Connection`, která je použita pro tvorbu dotazů v ostatních třídách databázové vrstvy.
- `DBIndex` - obsahuje metodu s názvem `selectLastWeatherTableData`, která vrací celou tabulku `LastWeather` (viz kapitola 6.3). Dále obsahuje metodu s názvem `selectServerInfoTableData`, která spouští proceduru s názvem `table_server_info` (viz kapitola 6.4) a vrací její výsledek.
- `DBPoruba2G`, `DBPoruba3G`, `DBBukovec2G` a `DBBukovec3G` - jsou hlavní třídy databázové vrstvy obsluhující vkládání a výběr dat z jednotlivých stanic. Obsahují metodu `insertInto`, která vkládá nově příchozí data z měřicích stanic a zároveň spouští procedury pro tvorbu hodinových a denních průměrů. Dále obsahuje spoustu metod pro výběr meteorologických dat a dat z mobilní sítě. Tyto metody obsahují buď nový dotaz na databázi (viz příklad 6.3), nebo spouštějí některou z již vytvořených uložených procedur (viz kapitola 6.4). Vstupem do těchto metod je vždy časové rozmezí výběru a mód, který udává, z jakého typu tabulky data vybírat (surová data, hodinový nebo denní průměr). Výsledky metod jsou použity jak pro výsledné grafy, tak i pro výsledné tabulky na webu.

```
...
ArrayList<Double> data = new ArrayList<Double>();
Connection conn = null;
try{
    conn = DBConnectionManager.getConnection();
    if (conn == null) {return data;}
    String query = "SELECT temperature FROM PorubaA2G
    WHERE datetime >= ? AND datetime <= ?;";

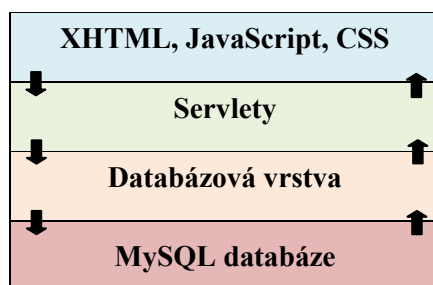
    PreparedStatement ps = conn.prepareStatement(query);
    ps.setLong(1, dateFrom);
    ps.setLong(2, dateTo);
    ResultSet result = ps.executeQuery();
    while (result.next())
    {data.add(result.getDouble(1));}
    }catch (SQLException ex){...}
    finally{
    if (conn != null){DBConnectionManager.closeConnection(conn);}
    }
    ...
```

---

Příklad 6.3: Ukázka dotazu do databáze v jazyce Java

## 7 Vizualizace dat

Webovou aplikaci lze rozdělit do několika vrstev (viz obrázek 7.1). Samotná vizualizace dat patří do vrstvy nejvyšší. Než se však data dostanou z MySQL databáze do samotných grafů, musí projít dvěma dalšími vrstvami - databázovou vrstvou, která byla popsána v kapitole 6.6, a vrstvou tvořenou Servlety. Poté se již tvoří grafy a tabulky pomocí JavaScriptu a vizualizační knihovny nazvané Google Charts. Tyto grafy a tabulky jsou následně umístěny v klasické XHTML stránce.



Obrázek 7.1: Jednotlivé vrstvy webové aplikace

### 7.1 Servlety

Servlety v této diplomové práci tvoří určité prostředníky mezi databázovou vrstvou (viz kapitola 6.6) a webovým rozhraním. Jinak řečeno, přetváří data získaná v datové vrstvě do podoby čitelné pomocí vizualizační knihovny Google Charts. Pro tyto účely má tato knihovna v jazyce Java svou speciální třídu nazvanou `DataSourceServlet`, která obsahuje jedinou metodu nazvanou `generateDataTable`. Tato metoda vrací instanci speciální třídy `DataTable`, která předává svůj obsah vizualizační knihovně ve formátu JSON. Servlety jsou dostupné pod určitou URL adresou. Pokud tedy zadáme správnou URL konkrétního Servletu, zobrazí se v prohlížeči jeho požadovaná odpověď (viz příklad 7.1). [9]

```
google.visualization.Query.setResponse(
{
  "version": "0.6",
  "status": "ok",
  "sig": "112171570",
  "table": {
    "cols": [
      { "id": "name", "label": "Type", "type": "string", "pattern": "" },
      { "id": "p3", "label": "Size", "type": "string", "pattern": "" }
    ],
    "rows": [
      { "c": [ { "v": "Available processors (cores):", "v": "8" } ] },
      { "c": [ { "v": "Free RAM (MB):", "v": "7117" } ] },
      { "c": [ { "v": "Maximum RAM (MB):", "v": "7855" } ] },
      { "c": [ { "v": "Total RAM available to JVM (MB):", "v": "7855" } ] },
      { "c": [ { "v": "Total HDD space (MB):", "v": "34141" } ] },
      { "c": [ { "v": "Free HDD space (MB):", "v": "16564" } ] },
      { "c": [ { "v": "Usable HDD space (MB):", "v": "14808" } ] }
    ]
  }
};
```

Příklad 7.1: Odpověď Servletu s URL `http://comtech-meteo.vsb.cz/index?value=hwinfo`

**Popis jednotlivých Servletů:**

- ServletPoruba2G, ServletPoruba3G, ServletBukovec2G a ServletBukovec3G - obsluhují veškeré grafy a tabulky na stránkách *poruba2G.xhtml*, *poruba3G.xhtml*, *bukovec2G.xhtml* a *bukovec3G.xhtml*, ale i grafy na stránce *index.xhtml*. K URL jednotlivých Servletů je zde přidán tzv. „*query string*“, který specifikuje požadovaná data (viz příklad 7.2). Jeden z parametrů „*query stringu*“ nazvaný *gap* určuje požadované časové období (0 pro poslední 1 hodinu, 1 pro 12 hodin, 2 pro 24 hodin, atd.). Pokud parametr *gap* obsahuje hodnotu znamenající vlastní časové rozmezí, jsou zde navíc ještě parametry *from* a *to* určující počátek a konec časového rozmezí. Další parameter nazvaný *value* určuje typ dat (např. *temperature*, *humidity*, atd.). Podle parametru *value* se volá konkrétní metoda z databázové vrstvy a vytváří požadovaný formát datové tabulky pro vizualizační knihovnu.
- ServletIndex - obsluhuje grafy a tabulky na hlavní stránce *index.xhtml* a tabulky na stránce *server.xhtml*. Princip je zde stejný jako v předchozím bodě.

---

<http://comtech-meteo.vsb.cz/poruba2G?gap=0&value=temperature>

---

Příklad 7.2: Ukázka URL požadující hodnoty teploty za poslední hodinu

## 7.2 Google Charts

Knihovna Google Charts je stvořena přímo pro účely vizualizace dat na webové stránce. Nabízí velký počet grafů různých typů. Nejběžnější způsob použití této knihovny je dohromady s jednoduchým JavaScriptem přidaným na webové stránce. Princip je takový, že se nejprve nahraje knihovna pro požadovaný graf, vytvoří se nebo nahrajou se data pro vykreslení, nastaví se vzhled a vlastnosti grafu a nakonec se vytvoří objekt grafu se specifickým ID. Na webové stránce se vytvoří tag `<div>` se stejným ID, který zobrazí vytvořený graf. Grafy jsou vykreslovány pomocí technologie HTML5/SVG, která zajišťuje kompatibilitu skrze prohlížeče a která umožňuje funkčnost na dalších platformách, jako iOS a Android.[12]

V této webové aplikaci se data naplňující grafy netvoří přímo v JavaScriptu, nýbrž pomocí již zmíněných Servletů popsaných v kapitole 7.1. Tyto Servlety se volají v JavaScriptové funkci pomocí speciální metody *Query*, která je pro tyto účely určená (viz příklad 7.3).

---

```
var poruba2G = new  
google.visualization.Query('poruba2G?gap=0&value=power');  
power2G.send(power2GResponse);
```

---

Příklad 7.3: Volání Servletu z JavaScriptové funkce a odeslání odpovědi další funkci

Pro každou odpověď ze Servletu je v JavaScriptu další funkce, která ji zpracovává (viz příklad 7.4). Tato funkce nejprve kontroluje, zda se nevyskytla nějaká chyba. Pokud je odpověď v pořádku, tak vytváří proměnnou obsahující nastavení vzhledu a vlastností daného grafu. Dále vybírá z odpovědi datovou tabulku ve formátu JSON. Nakonec vytváří objekt grafu podle ID a zavolá na něm metodu `draw`, kde vstupní parametry této metody tvoří právě data ve formátu JSON a vytvořené nastavení grafu. Pokud odpověď není v pořádku, nevykreslí se nic.

---

```
function power2GResponse(response) {
  if (response.isError()) { return; }
  var options = {
    axisTitlesPosition:'out',
    chartArea: {width:'85%',height:'68%'},
    colors:['#005aff','#00b6cd','#80ffc0','#67d29d','#4d9973','#9c007b','#
e7005a','#ff388c','#ff986a','#e0e778'],
    crosshair: { trigger: 'both',focused: { opacity: '0.5' },selected: {
opacity: '0.9' }},
    height:'300',
    interpolateNulls:true,
    legend: {position: 'bottom'},
    selectionMode: 'multiple',
    series: {
      0: {targetAxisIndex: '0'},
      1: {targetAxisIndex: '1'}},
    title:'The level of received signal strength by CellID',
    tooltip: { trigger: 'selection' },
    vAxis:{format:'##',gridlines:{color:
'#ccc',count:7},minorGridlines:{color: '#eee',count:1}},
    vAxes: {
      0: {title:'(dBm)',viewWindow:{max:'-50',min:'-110'}},
      1: {title:'',viewWindow:{max:'-50',min:'-110'}}}};
  var data = response.getDataTable();
  var chart = new google.visualization.LineChart(
    document.getElementById('chart_power2G'));
  chart.draw(data, options);};
```

---

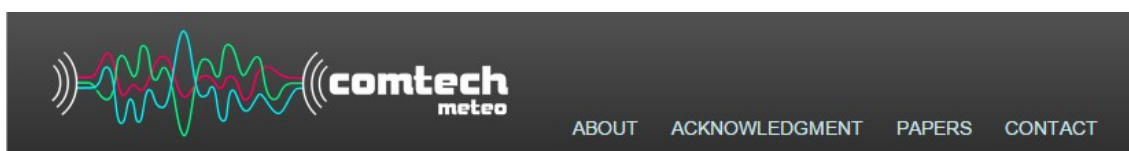
*Příklad 7.4: Metoda obsluhující odpověď ze Servletu a vykreslující graf*

### 7.3 XHTML stránky

Uživatelské rozhraní webové aplikace tvoří dohromady šest hlavních XHTML stránek. Jsou to *index.xhtml*, *server.xhtml*, *poruba2G.xhtml*, *poruba3G.xhtml*, *bukovec2G.xhtml* a *bukovec3G.xhtml*. Všechny tyto stránky však mají určité společné části, které se od sebe nijak neliší, a není tedy třeba je psát pokaždé zvlášť. Namísto toho jsou tyto komponenty vytvořeny v oddělených XHTML souborech pouze jednou. Komponenty musí být obaleny speciálním tagem `<ui:composition>` z technologie JSF. Do hlavní stránky je lze poté zakomponovat pomocí speciálního tagu `<ui:include>`. Výhoda tohoto způsobu je také v tom, že tuto komponentu stačí změnit pouze jednou a změny se projeví na všech hlavních stránkách.

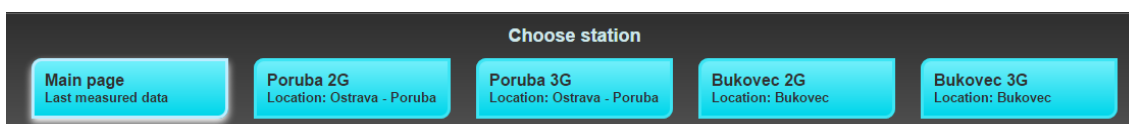
#### Popis jednotlivých komponent:

- *header.xhtml* - obsahuje hlavičku webové aplikace (viz obrázek 7.2). V této komponentě se nachází logo aplikace a určité informace o projektu. Tyto informace jsou v základu skryty. Zobrazují se jako dialogová okna pouze tehdy, pokud se klikne na jeden z odkazů v hlavičce. Tuto funkcionalitu zařizují JavaScriptové funkce s názvem `showPopup` a `hidePopup`.



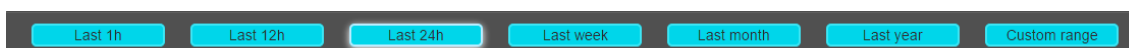
Obrázek 7.2: Hlavička webové aplikace

- *menubar.xhtml* - obsahuje hlavní menu (viz obrázek 7.3), které odkazuje na stránky *index.xhtml*, *poruba2G.xhtml*, *poruba3G.xhtml*, *bukovec2G.xhtml* a *bukovec3G.xhtml*. Pokud je zrovna zobrazena určitá stránka, je konkrétní tlačítko stránky označeno.



Obrázek 7.3: Hlavní menu webové aplikace

- *submenubar.xhtml* - obsahuje menu pro výběr časového rozmezí pro vykreslení (viz obrázek 7.4). Po kliknutí na jednu z položek se volá příslušná funkce v JavaScriptu a tlačítko se označí. Toto menu je zobrazeno pouze na stránkách pro jednotlivé stanice.



Obrázek 7.4: Menu pro výběr časového rozmezí

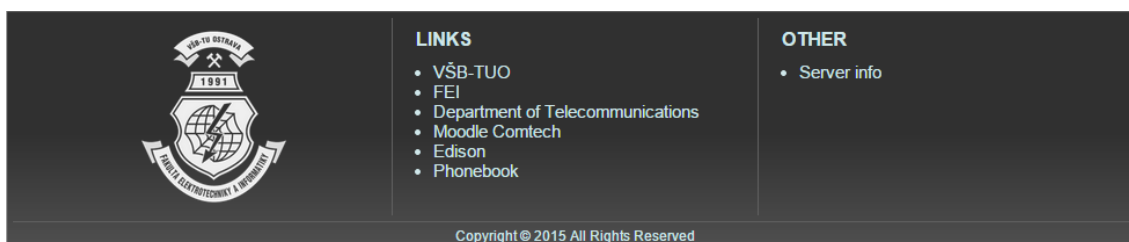
- *submenu\_custom.xhtml* - obsahuje položky pro výběr vlastního časového rozmezí (viz obrázek 7.5). Tato komponenta se objeví, pokud se v menu pro výběr časového rozmezí (viz obrázek 7.4) klikne na tlačítko s názvem „Custom range”.



Po vyplnění položek a kliknutí na tlačítko s názvem „*Confirm*” se volá funkce z JavaScriptu. Tato funkce zkontroluje vyplněné položky a podle nich nastaví „*query stringy*” při volání Servletů. Pokud zůstane některá z povinných položek nevyplněná, vyskočí chybová hláška a chybné políčko se označí.

Obrázek 7.5: Výběr vlastního časového období

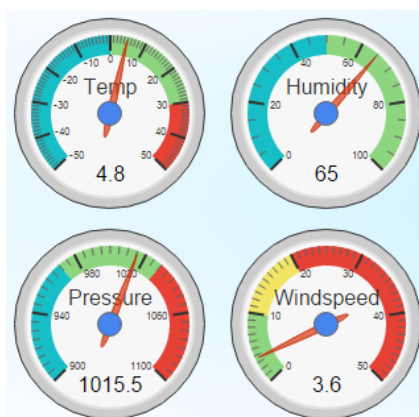
- *footer.xhtml* - obsahuje záhlaví webové aplikace (viz obrázek 7.6). V této komponentě se nachází logo fakulty, odkazy související s VŠB - TUO a v neposlední řadě odkaz na stránku *server.xhtml*.



Obrázek 7.6: Záhlaví webové aplikace

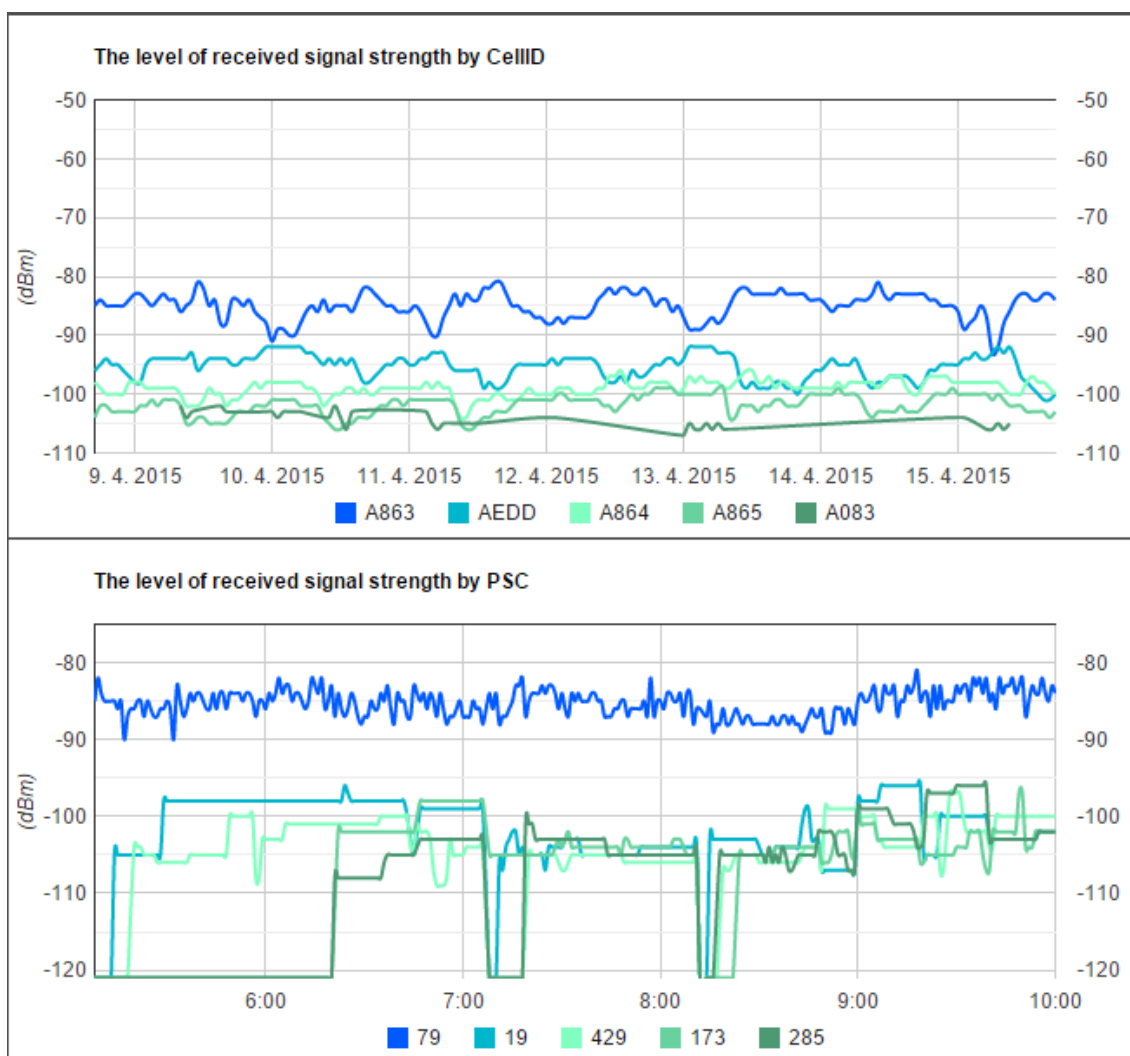
#### Popis hlavních stránek:

- *index.xhtml* - je úvodní stránka webové aplikace a zobrazuje poslední naměřené hodnoty ze všech měřících stanic. V první části obsahuje indikátory pro teplotu, vlhkost, tlak a rychlost větru (viz obrázek 7.7). Tyto čtyři indikátory jsou zde pro každou měřící stanici. V další části obsahuje tabulku posledních naměřených meteorologických hodnot pro všechny čtyři měřící stanice. Tabulka obsahuje název stanice, čas měření, teplotu, rosný bod, pocitovou teplotu, vlhkost, tlak, rychlost větru a směr větru. V poslední části jsou grafy pro každou stanici zobrazující poslední hodinu naměřené úrovně přijatého signálu.

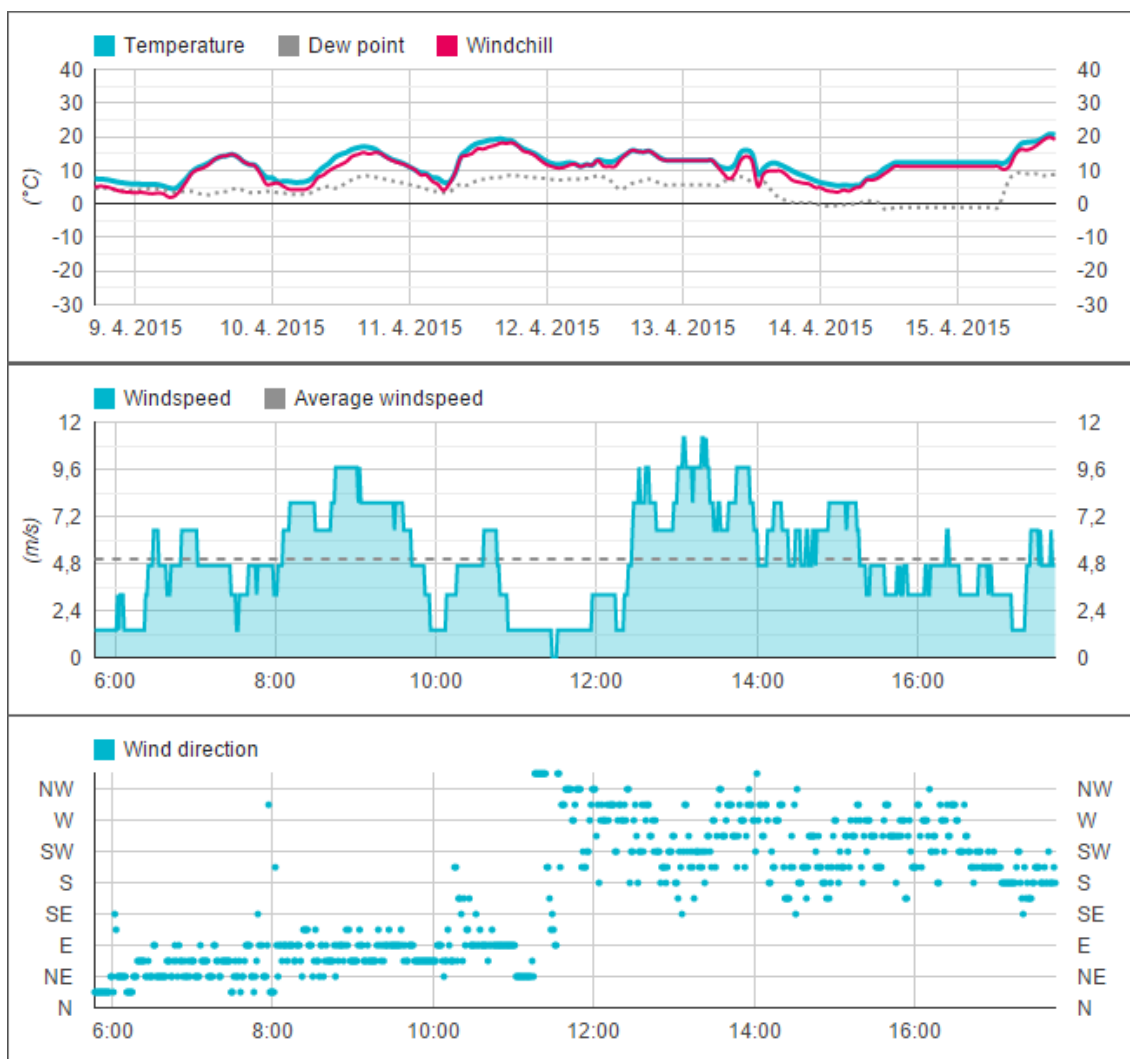


Obrázek 7.7: Indikátory teploty, vlhkosti, tlaku a rychlosti větru

- *poruba2G.xhtml*, *poruba3G.xhtml*, *bukovec2G.xhtml* a *bukovec3G.xhtml* - jsou stránky jednotlivých měřicích stanic. Složení všech těchto stránek je téměř shodné. Liší se pouze v grafu a tabulce pro naměřenou úroveň přijatého signálu, kde u *poruba2G.xhtml* a *bukovec2G.xhtml* jsou zobrazeny hodnoty z technologie GSM a u *poruba3G.xhtml* a *bukovec3G.xhtml* jsou zobrazeny hodnoty z technologie UMTS (viz obrázek 7.8). U technologie GSM se do grafu vykresluje úroveň přijatého signálu rozdělená podle paramteru CI. V tabulce je pak CI, LAC, BSIC, ARFCN a vybrané statistické metody pro úroveň přijatého signálu (maximum, minimum, průměr, medián, modus a standartní odchylka). U UMTS se do grafu vykresluje úroveň přijatého signálu rozdělená podle paramteru PSC. V tabulce je pak PSC a stejné statistické metody jako u GSM pro RSCP a Ec/No. Grafy a tabulky pro meteorologická data (viz obrázek 7.9) jsou pak pro všechny tyto stránky vzhledově stejné. Z meteorologických dat se vykreslují: teplota, rosný bod, pocitová teplota, vlhkost, déšť, tlak, rychlost větru a směr větru.



Obrázek 7.8: Ukázka grafů zobrazující úroveň přijatého signálu GSM (nahore) a UMTS



Obrázek 7.9: Ukázka grafů zobrazující meteorologické hodnoty

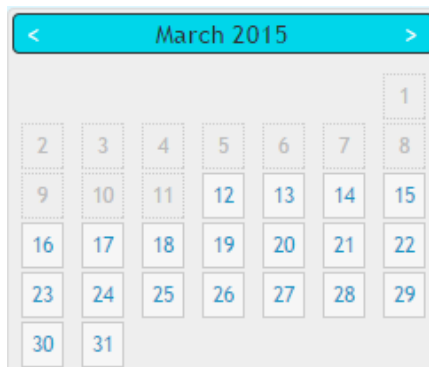
- *server.xhtml* - obsahuje pouze dvě tabulky. První tabulka udává informace o virtuálním stroji, tedy počet jader, informace o paměti RAM (velikost volné paměti a maximální velikost paměti) a o pevném disku (volné a celkové místo na disku). Druhá tabulka obsahuje informace o velikostech tabulek v MySQL databázi (viz obrázek 7.10).

Station	Row data [Size in MB]	Hour average [Size in MB]	Day average [Size in MB]
Poruba 2G	1458.83	37.08	2.88
Poruba 3G	1605.84	49.09	5.03
Bukovec 2G	659.66	13.03	0.58
Bukovec 3G	384.47	7.03	0.34

Obrázek 7.10: Tabulka udávající velikosti tabulek v MySQL databázi

## 7.4 Kalendář pro výběr data

V komponentě `submenu_custom.xhtml` je pro výběr data použit kalendář (viz obrázek 7.11), který je převzatý z [13]. Pro účely této diplomové práce je však jeho kód v JavaScriptu upraven. Byla přidána metoda, která omezuje maximální časového rozmezí mezi dvěma zde přítomnými kalendáři na 1 rok. Taktéž jsou upraveny barvy v souboru `datepickr.css`, který je určen pro stylování tohoto kalendáře.



Obrázek 7.11: *Kalendář pro výběr data*

## 7.5 Stylování

Pro stylování webové aplikace jsou použity klasické CSS styly. Jsou zde pouze dva takové CSS soubory. Již zmíněný soubor `datepickr.css`, který slouží pro stylování kalendáře, a soubor nazvaný `page.css`, který obsahuje veškeré zbývající třídy pro stylování webových stránek.

## 7.6 Logo webové aplikace

Logo pro webovou aplikaci je vytvořeno v grafickém nástroji Blender. Je vytvořeno ve dvou variantách. První varianta slouží pro zobrazení v hlavičce webové aplikace (viz obrázek 7.1) a druhá varianta slouží pro vystavení na jiné weby (viz obrázek 7.12).



Obrázek 7.12: *Logo aplikace*

## 8 Testování

Než se webová aplikace dostala na současný virtuální stroj, byla testována na předešlých dvou strojích. V prvotní fázi vývoje byla zkoušena na počítači, na kterém se současně i vyvíjelo. Zde byla vytvořena testovací databáze a server Apache Tomcat. Také se na tomto počítači spouštěly programy parsující soubory s naměřenými daty. Pokud zde bylo vše otestováno, přesunula se webová aplikace na testovací virtuální stroj, na kterém se vytvořila opět příslušná databáze a server. Programy parsující soubory byly již nasazeny na jednotlivé měřicí stanice. V tomto případě se ladila i průchodnost dat sítí. Nakonec byl vytvořen virtuální stroj určený pro finální nasazení webové aplikace. U programů na jednotlivých měřicích stanicích se změnila pouze cílová IP adresa. Nutno podotknout, že se objevily drobné chyby i za běhu na finálním stroji, které musely být opraveny jak na straně webové aplikace, tak i na straně měřicích stanic.

Z hlediska jednotlivých komponent celého systému, bylo testování rozděleno na několik kroků. Nejprve se testovaly programy parsující soubory s naměřenými daty. Poté se testovala webová služba a její komunikace s parsujícími programy. Dále následovalo testování ukládání dat do databáze a načítání dat databázovou vrstvou. Ve finále se testovaly Servlety a vizualizace jednotlivých dat na webové stránce.

Co se týká kompatibility, byla webová aplikace vyzkoušena v prohlížečích Google Chrome, Mozilla Firefox, Internet Explorer 11 a Safari. Z hlediska platforem pak byla otestována na iOS a Androidu. Ve všech těchto případech funguje webová aplikace spolehlivě.

## Závěr

V rámci teoretické části této diplomové práce jsem popsal měření parametrů rádiového kanálu u systému GSM a UMTS. Uvedl jsem konkrétní měřené parametry, dále za jakým účelem, jakým způsobem a na jakých kanálech se tyto parametry měří.

V praktické části jsem vytvořil automatizovaný systém, který navazuje na již stávající řešení. Vytvořil jsem programy, které zpracovávají každou minutu soubory s naměřenými daty a které tato zpracovaná data posílají na server skrze webové služby. Dále jsem navrhl strukturu databáze, do které se přijatá data ukládají, a vytvořil optimalizovanou webovou aplikaci vizualizující uložená data z databáze v grafech a tabulkách pro různé časové období. Nakonec jsem webovou aplikaci otestoval.

Automatizovaný systém je navržen tak, aby byl schopný bez lidského zásahu fungovat minimálně dalších 10 let. Tato doba je daná především velikostí pevného úložiště na virtuálním stroji. Do budoucna by se dal tento systém rozšířit o další měřicí stanice nebo dokonce spojit i s jinou diplomovou prací, která má za úkol měřit hodnoty optického bezvláknového spoje v závislosti na počasí. Dále by bylo možné přidat více statistických metod. Data uložená v databázi jsou také vhodná pro zadání další diplomové práce, která by mohla tato data analyzovat z jiného pohledu a hledat různé anomálie nebo závislosti úrovně přijatého signálu na počasí.

Vytvořená webová aplikace bude sloužit široké veřejnosti, která bude mít o tuto problematiku zájem. Její URL je <http://comtech-meteo.vsb.cz/>. Odkaz na ní je také uveden na stránkách Katedry telekomunikační techniky Vysoké školy báňské - Technické univerzity Ostrava.

Prostřednictvím této diplomové práce jsem měl možnost seznámit se s různými, pro mne dosud neznámými technologiemi, což pro mne bylo velkým přínosem. Před zahájením této práce jsem neznal webové služby, dále jsem nikdy nepracoval s MySQL databází, serverem Apache Tomcat, JavaScriptem a knihovnou Google Charts. Pouze okrajově jsem použil Java Servlety a CSS styly. Nyní jsou mé znalosti rozšířeny o všechny tyto technologie a také jsem získal nové zkušenosti, co se týče jejich používání.

## Použitá literatura a zdroje

- [1] EBERSPÄCHER, Jörg, Hans-Jörg VÖGEL a Christian BETTSTETTER. *GSM switching, services and protocols*. 2nd ed. Chichester [England]: Wiley, 2001. ISBN 04-708-4174-5.
- [2] ROMERO, Javier, Timo HALONEN a Juan MELERO. *GSM, GPRS and EDGE Performance Evolution Towards 3G/UMTS*. 2. vyd. Chichester: John Wiley, 2004. ISBN 978-047-0866-955.
- [3] SAUTER, Martin. *From GSM to LTE: An introduction to mobile networks and mobile broadband*. Chichester, West Sussex, U.K.: Wiley, 2011. ISBN 04-709-7822-8.
- [4] TOSKALA, Antti a Harri HOLMA. *WCDMA for UMTS: HSPA evolution and LTE*. 4th ed. Editor Harri Holma, Antti Toskala. Hoboken, NJ: Wiley, 2007, xxxi, 539 p. ISBN 978-047-0870-969.
- [5] RAHNEMA, Moe. *UMTS network planning, optimization, and inter-operation with GSM*. Singapore: John Wiley, 2008, xvii, 327 s. ISBN 978-0-470-82301-9.
- [6] *Radio network planning and optimisation for UMTS*. 2nd ed. Editor Jaana Laiho, Achim Wacker, Tomáš Novosad. Chichester: Wiley, 2006, xxxi, 629 s. ISBN 04-700-1575-6.
- [7] TIDWELL, Doug, James SNELL a Pavel KULCHENKO. *Programming Web services with SOAP*. 1st ed. Sebastopol, CA: O'Reilly, 2002. ISBN 978-059-6552-015.
- [8] SCHNEIDER, Robert D. *MySQL: oficiální průvodce tvorbou, správou a laděním databází*. Praha: Grada Publishing, 2006, 372 s. ISBN 80-247-1516-3.
- [9] HUNTER, Jason a William CRAWFORD. *Java Servlet Programming*. 2nd ed. Sebastopol: O'Reilly Media, Inc, 2010. ISBN 978-144-9390-679.
- [10] *Mojarra JavaServer Faces* [online]. 2015 [cit. 2015-04-05].  
Dostupné z: <https://jaserverfaces.java.net/>
- [11] *Apache CXF* [online]. 2015 [cit. 2015-04-05].  
Dostupné z: <http://cxf.apache.org/index.html>
- [12] *Google Charts: Using Google Charts* [online]. 2015 [cit. 2015-04-02].  
Dostupné z: <https://developers.google.com/chart/interactive/docs/index>
- [13] *Datepickr* [online]. 2014 [cit. 2015-04-02].  
Dostupné z: <https://github.com/joshsalverda/datepickr>

## Přílohy

Součástí diplomové práce je CD.

Adresářová struktura přiloženého CD:

Název složky	Popis
Programy	Jednotlivé zkompilevané programy
Vypracovani	Vypracování DP v elektronické podobě
Zdrojove_kody	Zdrojové kódy všech programů